

Implementace portu komunikační knihovny pro LoRaWAN pro energeticky nezávislé zařízení IoT

Implementation of Port of a LoRaWAN Communication Stack for Maintenance

Free IoT Device

Bc. Lukáš Granzer

Diplomová práce

Vedoucí práce: Ing. Jaromír Konečný, Ph.D.

Ostrava, 2021

Abstrakt

Tato diplomová práce se zabývá problematikou energeticky nezávislých zařízení a nízkoenergetických bezdrátových sítí. Práce shrnuje klíčové parametry vybraných nízkoenergetických bezdrátových technologií, které jsou v současnosti na trhu. V dalších kapitolách se věnuje pouze síti LoRaWAN a možnostem využití volně dostupných softwarových knihoven pro vestavěná zařízení. V praktické části práce se zabývám portací oficiální knihovny síťového stacku od LoRa Alliance na mikrokontrolér z rodiny Cortex M0+ FRDMKL25Z, pro který nebyla tato knihovna původně určena. Cílem je navázání rádiového spojení s využitím této knihovny.

Klíčová slova

LPWAN, LoRaWAN, Sigfox, IQRF, NB-IoT, LTE-M, síťový stack, nízkoenergetická zařízení

Abstract

The diploma thesis deals with various aspects of energetic independent devices and low-power wireless networks. The summarization of key features was written for low-power wireless technologies available on the market these days. The next chapters are mainly focused on LoRaWAN and options in choosing open-source software libraries for embedded devices. The practical part covers portation of the official version network stack library released by LoRa Alliance association. The target platform is a microcontroller Cortex M0+ FRDMKL25Z, which was not originally supported with this library. The goal is establish radio connection using this library.

Keywords

LPWAN, LoRaWAN, Sigfox, IQRF, NB-IoT, LTE-M, network stack, low-power devices

Poděkování

Rád bych tímto poděkoval vedoucímu práce Ing. Jaromíru Konečnému, Ph.D. za všechny věnovaný čas a konzultantovi Ing. Martinu Stankušovi, Ph.D. za ochotnou pomoc s řešením všech problémů, které se vyskytly. Další poděkování patří Ing. Radovanu Háčovskému, Ph.D. za poskytnutí cenných rad a zapůjčení laboratorních přístrojů, bez kterých by nebylo možné tuto práci dokončit.

Především děkuji mé rodině, přátelům, a zejména Bohu, že mi byli po celou dobu studia oporou a podporovali mě.

Obsah

Seznam použitých symbolů a zkratk	6
Seznam obrázků	8
Seznam tabulek	10
1 Úvod	11
2 Energeticky nezávislá zařízení IoT	13
2.1 Primární baterie a akumulátory	15
2.2 Alternativní zdroje energie	17
3 Bezdrátové sítě typu LPWAN	21
3.1 LoRaWAN	22
3.2 IQRf	29
3.3 Sigfox	36
3.4 NB-IoT	39
3.5 LTE-M	40
4 Možnosti implementace komunikačního rozhraní	43
4.1 IBM LMIC framework	44
4.2 Arduino-LMIC	46
4.3 LoRa MAC node	47
5 Portace softwarové knihovny sítě LoRaWAN na vybraný mikrokontrolér	49
5.1 Vývojový kit FRDM-KL25Z	50
5.2 Modem s SX1261	52
5.3 Vnitřní struktura knihovny LoRa MAC node	53
5.4 Provedené úpravy	54
5.5 Obsluha GPIO	54
5.6 Sériová komunikace	55

5.7	Plánovač úloh	55
6	Testování softwaru a zkušební přenos dat	59
6.1	Ovládání GPIO	59
6.2	SPI komunikace	59
6.3	UART komunikace	60
6.4	Zkouška časování	60
6.5	Plánovač událostí	60
6.6	Test LED indikátorů	61
6.7	Komunikace s modemem	61
6.8	Zkušební spojení s LoRaWAN bránou	61
7	Zhodnocení výsledků	65
8	Závěr	67
	Literatura	68

Seznam použitých zkratek a symbolů

ABP	– Activation by personalization
DBPSK	– Differential binary phase shift keying
DMA	– Direct memory access
DPA	– Direct peripheral access
DSSS	– Direct sequence spread spektrum
EEPROM	– Electrically erasable programmable read-only memory
FDMA	– Frequency division multiple access
GFSK	– Gaussian frequency shift keying
GPIO	– General purpose input/output
HAL	– Hardware abstraction layer
IoT	– Internet of things
LMIC	– LoRa MAC in C
LoRa	– Long range
LoRaWAN	– Long range wide area network
LPP	– Low power payload
LPWAN	– Low-power wide-area network
MAC	– Medium access control
M2M	– Machine to machine
OFDMA	– Orthogonal frequency-division multiple access
OS	– Operační systém
OTAA	– Over-the-air activation
USB OTG	– Universal serial bus on-the-go
PIT	– Periodical interrupt timer
PWM	– Pulse width modulation
QPSK	– Quadrature phase-shift keying
SPI	– Serial peripheral interface
SWD	– Serial wire debug
TDMA	– Time division multiple access

UART	– Universal asynchronous receiver-transmitter
UNB	– Ultra narrow band
VRN	– Virtual routing number
VoLTE	– Voice over LTE

Seznam obrázků

2.1	Relativní poměr spotřeby energie v embedded systémech [4]	14
2.2	Proudový odběr senzoru TMP102 v průběhu měření. Frekvence měření 4 Hz. [6]	15
2.3	Porovnání energetické hustoty primárních baterií a akumulátorů.	16
2.4	Moderní provedení hodin typu Atmos	18
3.1	Rekordní přenos na vzdálenost 702 km [16]	23
3.2	Aktivita zařízení při úspěšném příjmu v oknu Rx1	26
3.3	Aktivita zařízení při úspěšném příjmu v oknu Rx2	26
3.4	Průběh komunikace v třídě B	27
3.5	Průběh komunikace v třídě C	27
3.6	LoRaWAN síť v režiji TheThingsNetwork 3/2021 [18]	28
3.7	LoRaWAN síť v režiji Českých radiokomunikací 6/2020 [19]	29
3.8	Rádiový modul IQRF	30
3.9	Struktura aplikace bez použití DPA [23]	35
3.10	Struktura aplikace s DPA [24]	35
3.11	Pokrytí Evropy sítí Sigfox v roce 2021 (modře), ve výstavbě (fialově) [26]	36
3.12	Pokrytí sítí NB-IoT [35]	40
3.13	Pokrytí sítí LTE-M [35]	42
4.1	Matthijs Kooijman	44
4.2	Modem osazený čipem Semtech SX1276	46
4.3	Partneři projektu	47
5.1	Vývojová deska FRDM-KL25Z	50
5.2	Blokový diagram debuggeru	51
5.3	Modem SX1261	52
5.4	Datová struktura pro plánovač	56
5.5	Přidání nové úlohy do plánovače	57
5.6	Obsluha přerušení PIT	58

6.1	Trasa rádiového spoje s bránou rrz-pomnikowa24	62
6.2	Výpis komunikace sítě LoRaWAN	63
6.3	Trasa a výškový profil rádiového spoje	64
6.4	Výpis dat z backendu LoRaWAN	64

Seznam tabulek

2.1	Porovnání spotřeby různých typů low-power mikrokontrolérů [2, 3]	14
2.2	Ukázka rozdílu ve spotřebě senzorů pro low-power aplikace	15
2.3	Přehled parametrů primárních článků [4]	16
2.4	Předpokládané samovybíjení baterií dle typu [9]	17
2.5	Měření intenzity RF signálu v různých prostředích [11]	19
3.1	Možné faktory šíření pro šířku pásma 125 kHz [17]	24
3.2	Přibližná doba potřebná ke sběru dat ze všech uzlů v síti	33
3.3	Ukázka ceníku Sigfoxu pro ČR [32]	38
5.1	Možnosti napájení vývojové desky	51
5.2	Doporučené provozní podmínky modemu SX1261	53
5.3	Mezní provozní hodnoty modemu SX1261	53

Kapitola 1

Úvod

S rozvojem techniky a automatizace ve 21. století vzrůstá potřeba zasílat zprávy na velké vzdálenosti. Technický prostředek, který by umožňoval plnohodnotně přenášet zprávy na dlouhé vzdálenosti, byl dlouhou dobu nedostupnou záležitostí. Pokud pomineme holubí poštu, lze za první plnohodnotný mechanický komunikační prostředek považovat Chappův telegraf. Jednalo se o optický telegrafní systém na bázi signálních ramen. V různých obměnách se následně rozšířil po celé Evropě. Na spojnících velkých měst se postupně budovaly signální věže, které propojily jednotlivé evropské metropole. Další obrat nastal až s vynálezem kabelového telegrafu v roce 1838 Samuelem Morseem. Na rozdíl od většiny tehdejších metod dálkové komunikace, nevyžadoval tento způsob přímou viditelnost mezi odesílatelem a příjemcem zprávy. Přenos zpráv, co do rychlosti, tak poprvé předčil po dlouhou dobu nejrychlejší komunikační prostředek, holubí poštu. Stále se ale nejednalo o přenosné zařízení, takže jeho použití bylo limitováno na stacionární aplikace. [1]

Nyní již uběhlo více než 100 let od okamžiku, kdy byly objeveny rádiové vlny a Guglielmo Marconi vynalezl bezdrátový telegraf. V začátcích byly možnosti rádiového přenosu značně omezené, ale i tak se jednalo o průlomovou technologii na poli dálkové komunikace. Za další významný milník lze považovat vynález amplitudové modulace z roku 1900. Poprvé tak bylo možné přenášet lidský hlas. Tím byla odstartována éra modulovaných rádiových přenosů tak, jak je známe dnes. Následný přechod k digitálním modulacím už byl pouze dalším vývojovým stupněm této technologie. Využití digitálních modulací v rádiové technice se budu více věnovat v kapitole Bezdrátových sítí typu LPWAN. Zkratkou LPWAN¹ zároveň narážíme na další aspekt rádiového vysílání, a tím je energetická náročnost samotného rádiového vysílání. Propojením rádiových technologií a embedded zařízení dostáváme typické zařízení IoT – Internetu věcí. Právě u tohoto typu elektroniky hraje energetická náročnost hlavní roli, protože zásadně ovlivňuje konstrukci a častokrát i životnost daného zařízení. V 2. kapitole proto najdete stručný přehled problematiky energeticky nezávislých zařízení. Jsou zde srovnány různé typy mikrokontrolérů a senzorů z hlediska jejich spotřeby elektrické energie. Dále je zde zpracován přehled různých bateriových technologií s důrazem na dlouhodobý provoz

¹Low power wide area network – nízkoenergetické telekomunikační síť

zařízení s malou spotřebou energie. Následně porovnávám energetický potenciál různých technik pro získávání elektrické energie z okolního prostředí. Protože se ale jedná o rozsáhlé téma, které již bylo mnohokrát popsáno v jiných publikacích, nejsou zde uvedeny technické detaily jednotlivých řešení, ale spíše se zaměřuji na energetický potenciál vybraných zdrojů energie.

Další část práce se zaměřuje na implementaci komunikačního rozhraní sítě LoRaWAN s využitím open source knihoven pro mikrokontroléry. Shrnuji zde vývoj tří různých řešení a možnosti, které dané knihovny nabízejí. Jedna z knihoven je následně podrobněji popsána v praktické části práce.

Praktická část práce se zabývá problematikou portace oficiálního softwaru pro komunikaci pomocí sítě LoRaWAN. Cílovou platformou je vývojový kit FRDM-KL25Z a modem založený na mikročipu SX1261. Po úpravách bude knihovna otestována, aby se zjistilo, zda je tento software vůbec vhodný pro výše uvedené embedded zařízení. Následně bude pro demonstrační účely vytvořena jednoduchá vzorová aplikace. Z hlediska nízkenergetických zařízení nás zde bude zajímat především komunikační režim ve třídě B. Tato třída komunikace umožňuje obousměrnou výměnu dat mezi koncovým zařízením a bránou. Protože je v této třídě rádiové spojení založené na přesném načasování přenosových oken, dovoluje nám to v mezechasech vypínat koncové zařízení a ušetřit tak značné množství energie.

Kapitola 2

Energeticky nezávislá zařízení IoT

Každé elektrické zařízení potřebuje pro svoji činnost zdroj energie. U většiny zařízení je tento zdroj zajištěn prostým napojením na elektrickou distribuční síť. Můžeme tak čerpat energii v téměř neomezené míře za poměrně příznivých cenových podmínek. V případech, kde je elektrické zařízení v pohybu, už toto řešení naráží na své limity. Dominantním zdrojem pro tuto oblast elektrických zařízení se tak stávají elektrochemické zdroje elektrické energie. Tímto krokem ale automaticky vstupují do hry parametry jako je životnost, kapacita a energetická hustota zdroje. Přítomnost elektrochemického zdroje v zařízení navíc zvyšuje jeho hmotnost, což v případě pohyblivých zařízení s elektrickým pohonem vede ke zvýšení spotřeby elektrické energie, tím zase vzroste nárok na zdroj energie a tím se dostáváme do uzavřeného kruhu. Každé vyčerpání tohoto omezeného zdroje energie s sebou nese další dodatečné úkony potřebné k jeho výměně, případně dobíjení energie u akumulátorů. U některých průmyslových nebo podvodních aplikací je takováto výměna baterie navíc spojená se značnou pracností celé výměny, protože je potřeba provést složitou demontáž zařízení nebo vstoupit do nebezpečného prostředí.

Pečlivým návrhem zařízení je sice možné snížit jeho provozní spotřebu energie, ale i tak tím pouze prodlužujeme servisní interval zařízení. Problematika údržby a výměny baterií u velkého počtu zařízení byla do nedávné doby spíše záležitostí průmyslových aplikací. S rozmachem spotřební elektroniky už ale začíná tento problém pomalu zasahovat i do běžných domácností. V závislosti na délce servisního intervalu vždycky od určitého počtu bateriových zařízení narazíme na neudržitelnost tohoto řešení. Nemluvě o ekologických dopadech celé situace.

Typické elektrické měřicí zařízení s rádiovou konektivitou spotřebovává elektrickou energii několika činnostmi:

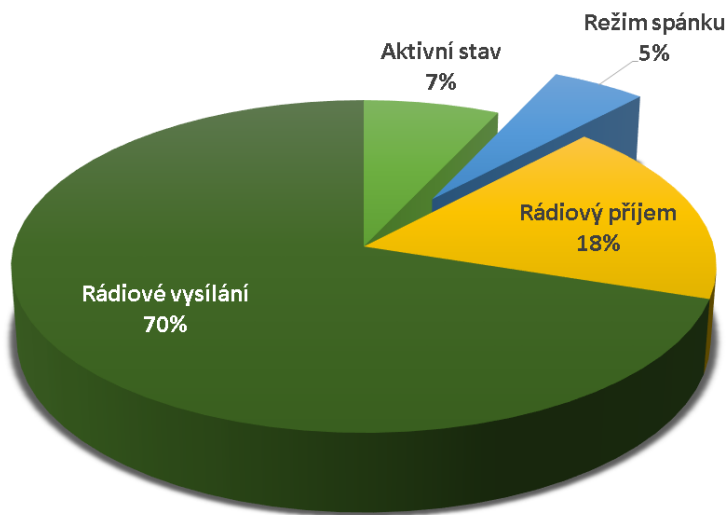
- Vykonáváním řídicího algoritmu.
- Sběrem informací pomocí senzorů.
- Komunikací s okolím.

Energetické nároky řídicího prvku se dají poměrně dobře ovlivnit vhodným návrhem elektrických obvodů a výběrem mikrokontroléru. U mikrokontroléru můžeme navíc omezit taktovací frekvenci na nezbytné minimum nebo v mezičase operací použít režim spánku. Pro názornost zde uvedu srovnání spotřeby vybraných typů mikrokontrolérů pro low-power aplikace.

Tabulka 2.1: Porovnání spotřeby různých typů low-power mikrokontrolérů [2, 3]

Mikrokontrolér	Architektura	Odebíraný proud		Rozsah napájecího napětí
		Provozní	Režim spánku	
PIC16F153x	8 bit	45 μA při 1 MHz	60 nA	1,80–5,5 V
ATtiny21x/41x	8 bit	600 μA při 1 MHz	100 nA	1,80–5,5 V
SAM L10	32 bit	25 $\mu\text{A}/\text{MHz}$	500 nA	1,62–3,6 V
SAM E5x	32 bit	70 $\mu\text{A}/\text{MHz}$	3,4 μA	1,62–3,6 V
SAM D5x	32 bit	65 $\mu\text{A}/\text{MHz}$	10 μA	1,71–3,6 V

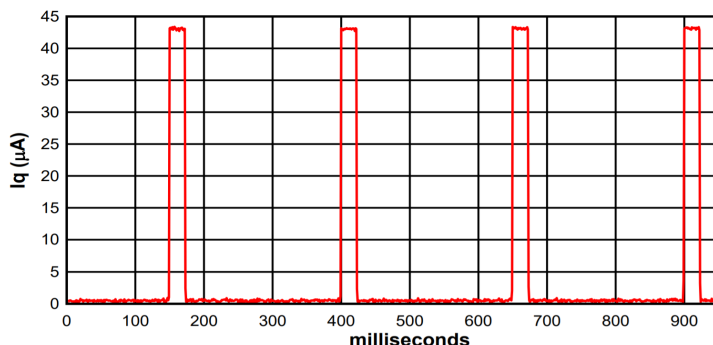
Největší podíl na celkové spotřebě zařízení má vždy rádiová komunikace. Použitím rádiových sítí typu LPWAN je sice možné tuto spotřebu o něco snížit, stále se ale bude jednat o proudy okolo desítek až stovek miliampér.



Obrázek 2.1: Relativní poměr spotřeby energie v embedded systémech [4]

Dalším významným spotřebičem může být u monitorovacích zařízení jejich senzorická část. Zde se spotřeba různých typů senzorů značně liší v závislosti na fyzikálním principu měření. U některých fyzikálních veličin se ale úspěšně daří minimalizovat spotřebu příslušných senzorů. Jako ukázkou bych zde uvedl digitální teplotní senzor TMP102 od společnosti Texas Instruments. Senzor má pracovní

rozsah -40 až $+125^{\circ}\text{C}$, s přesností po kalibraci $\pm 0,5^{\circ}\text{C}$. Vnitřní AD převodník má 12bitové rozlišení. Elektronika senzoru je přizpůsobena nízkoenergetickým aplikacím, takže je například možné v mezech vypínat vnitřní logiku senzoru. [5]



Obrázek 2.2: Proudový odběr senzoru TMP102 v průběhu měření. Frekvence měření 4 Hz. [6]

U chemických senzorů, jako jsou plynové detektory pro CO nebo uhlovodíky (methan, ethanol), je spotřeba obecně vyšší. Důvodem je častokrát nutnost ohřevu detekční části senzoru na teploty okolo 300°C . Dosažení takové teploty navíc někdy trvá i 60 sekund a po tuto celou dobu senzor spotřebovává zvýšené množství energie. U běžných typů senzorů může být proudový odběr 70 mA, nebo i 150 mA (MQ-7 – senzor CO a MQ-2 – senzor hořlavých plynů), u speciálních low-power typů na principu NDIR¹ se spotřeba pohybuje i okolo 1 mA. Výběr vhodného typu senzoru tedy může několikanásobně snížit energetickou náročnost měřicího zařízení a naopak.

Tabulka 2.2: Ukázka rozdílu ve spotřebě senzorů pro low-power aplikace

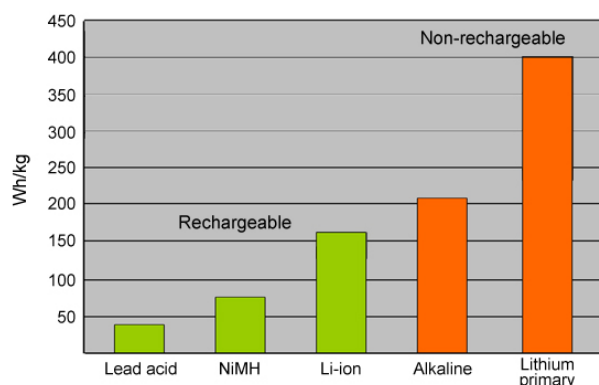
Veličina	Běžný senzor		Low-power senzor	
	Typ	Průměrná spotřeba	Typ	Průměrná spotřeba
Teplota a vlhkost	AM2302	300 μA	HDC2010	105–550 nA
Vibrace a pohyb	MPU-6000	10 μA –3,8 mA	LIS3DH	2–11 μA
Tlak	MPX2202	6 mA	SP300	530 μA

2.1 Primární baterie a akumulátory

Bateriová zařízení je možné obecně napájet buď primárními články nebo nabíjecími akumulátory. V současnosti se v elektrických zařízeních setkáváme spíše s akumulátory, nicméně v některých aplikacích je stále výhodnější použít primární baterii. Primární baterie mají několikanásobně vyšší

¹Non-dispersive infrared

energetickou hustotu, takže je možné dosáhnout příznivějšího poměru mezi velikostí zařízení a dobou provozu. Situaci ilustruje následující graf, převzatý z webu batteryuniversity.com.



Obrázek 2.3: Porovnání energetické hustoty primárních baterií a akumulátorů.

Dalším parametrem, ve kterém primární baterii překonávají nabíjecí články, je životnost článku v různých fázích životního cyklu baterie. Primární baterie mají velmi minimální samovybíjení, takže dosahují daleko větší skladovací životnosti než akumulátory. U některých typů nabíjecích akumulátorů se můžeme běžně setkat s případy, kdy po delší době skladování dorazí akumulátor v takovém stavu, že už vlastním samovybíjením došlo k jeho podbití a s tím spojené ztrátě kapacity. Tento problém se nepřímo promítá i do životnosti nízkoenergetických zařízení.

Tabulka 2.3: Přehled parametrů primárních článků [4]

Typ	Materiál elektrod	Energetická hustota (MJ/kg)	Napětí (V)	Doba skladování (roků)	Vnitřní odpor (mΩ)	Provozní teplota (°C)
Alkalická	Zn – MnO ₂	0,50	1,5	5–7	150–300	18 až 55
Zinko-uhlíková	Zn – MnO ₂	0,13	1,5	2-3		0 až 50
Lithiová (BR)	Li – CFx	1,30	3	7–10	1 000-4 000	-40 až 60
Lithiová (CR)	Li – MnO ₂	1,00	3	7–10	150–300	-40 až 60
Lithiová (Fe)	Li – FeS ₂	1,10	1,5	15	90–150	-40 až 60
Lithium thionyl chloridová	Li – SOCl ₂	1,04	3,6	20		-55 až 60
Zinkovzduchová	Zn – O ₂	1,69	1,4	4		0 až 50

Typicky v bezpečnostních aplikacích existuje spousta zařízení, která po většinu svojí životnosti neodebírají žádný proud. Teprve v reakci na vnější signál, např. zaplavení snímacích elektrod vodou nebo změna náklonu nebo náraz, aktivují elektroniku a začnou vysílat poplašný signál. Pro takovéto aplikace potom může být hlavním parametrem limitujícím životnost zařízení právě samovybíjení baterie. Ze stejného důvodu se pro dlouhodobé napájení zařízení nepoužívají superkondenzátory. Vlivem samovybíjení u nich totiž dochází k poklesu napětí z 100 % na 50 % za pouhých 30 až 40 dní. Problematické je i zužítkovat celý napěťový rozsah kondenzátoru, protože v poslední fázi vybíjení už nízké napětí nestačí ani k provozu zvyšujícího měniče, takže nikdy nevyužijeme celou kapacitu.

V případě použití nabíjecích článků musíme zase zařízení vybavit obvodem pro monitorování stavu akumulátoru. Vybití kteréhokoli článku akumulátoru pod určité kritické napětí vede k jeho degradaci, proto je žádoucí tomuto stavu zabránit a raději včas odpojit spotřebič. [7, 8]

Tabulka 2.4: Předpokládané samovybíjení baterií dle typu [9]

Typ baterie	Předpokládané samovybíjení
Primární lithiový článek	10 % za 5 let
Alkalický primární článek	2–3 % za 1 rok (předpokládaná životnost 7–10 let)
Pb akumulátor	5 % za měsíc
NiCd, NiFe	10–15 % první den, následně 10–15 % měsíčně
Lithiové akumulátory	5 % první den, následně 1–2 % měsíčně

2.2 Alternativní zdroje energie

U některých bateriových zařízení není možné zajistit dostatek elektrické energie pouhým použitím baterií. Buď by byl servisní interval příliš krátký, nebo by hmotnost akumulátoru neúměrně zvětšila rozměry zařízení. V jiných aplikacích zase narážíme na problémy s dostupností napájení, protože jsou vzdálenosti mezi senzory tak velké, že se nevyplácí instalovat napájecí kabely ke každému senzoru. V některých systémech se zase senzory mohou nacházet na rotujících částech strojů.

Častokrát se ale v místě měřicího zařízení nachází jiná, neelektrická forma energie. Může se jednat o teplotní rozdíl dvou částí konstrukce, proudící kapalinu, sluneční záření nebo silnou vibraci. Konverzí této energie na energii elektrickou tak můžeme získat potřebný lokální zdroj pro napájení našeho zařízení. Z historického hlediska se nejedná o žádnou novinku. Už v 17. století existovaly první snahy o zužítkování změn atmosférického tlaku a teploty k pohonu mechanických hodin. V roce 1928

postavil švýcarský inženýr Jean-León Reutter první plnohodnotný prototyp mechanických hodin, který byl poháněn čistě změnami okolní teploty. Tyto hodiny je dokonce možné koupit i dnes pod označením Atmos.



Obrázek 2.4: Moderní provedení hodin typu Atmos

V našich aplikacích nás ale bude spíše zajímat konverze okolních zdrojů na energii elektrickou. Pro tuto technologii se v současné technické praxi vžil anglický pojem *energy harvesting* – neboli získávání energie. Jedná se o poměrně široké spektrum technologií, protože každé prostředí má úplně jiné provozní podmínky pro získávání energie. Většina takovýchto zařízení je ale zároveň úzce zaměřená na konkrétní zdroj energie a určitou aplikaci, takže se málokdy jedná o univerzálně použitelný zdroj elektrické energie.

Zařízení pro *energy harvesting* můžeme rozdělit podle typu zdroje a principu transformace energie do čtyř kategorií.

2.2.1 Solární a světelné zdroje

Asi jedním z nejdostupnějších zdrojů energie je sluneční záření. Nabízí poměrně vysoký poměr mezi množstvím získané energie a potřebnou aktivní plochou. Určitým hendikepem je pravidelná ztráta tohoto zdroje během nočních hodin. Extrémním případem je pak nasazení této technologie v polárních oblastech, kde je potřeba vytvořit pro nepřetržitý provoz půlroční zásobu energie. Ve venkovním prostředí je možné dosáhnout zisku až 100 mW/cm^2 . O něco horší situace nastává v aplikacích uvnitř budov, kde jsme odkázáni na poměrně slabé světelné zdroje. Zde se dostáváme na hodnoty v rozsahu $10 \mu\text{W/cm}^2$ – 100 mW/cm^2 . Opět se ve většině případů jedná o přerušovaný zdroj energie, který je ovlivňovaný provozní dobou dané budovy. [10]

2.2.2 Termoelektrické generátory

Tyto zdroje využívají Seebeckova jevu k přeměně tepelné energie na elektrickou. Množství získané energie je závislé na teplotním gradientu. Bohužel má tento proces konverze poměrně nízkou efek-

tivitu, což je nutné kompenzovat větší plochou generátoru. Energetická hustota tohoto zdroje se pohybuje v rozmezí $10 \mu\text{W}/\text{cm}^2$ až $1 \text{ mW}/\text{cm}^2$. Jedná se o poměrně široce dostupný zdroj energie, protože většina energetických ztrát v mechanických i elektrických zařízeních se projevuje uvolněním odpadního tepla, které je možné takto znovu zužitkovat a konvertovat zpět na elektrickou energii. Další možnou oblastí využití termoelektrických generátorů je nositelná elektronika. Zde se využívá teplotního rozdílu mezi lidským tělem a okolním prostředím. [10]

2.2.3 Elektromagnetické zdroje

Napájení pomocí elektromagnetických polí můžeme rozdělit do dvou kategorií. První možností je napájení na krátkou vzdálenost pomocí elektromagnetické indukce a rezonance. Tento typ přenosu se vyznačuje poměrně vysokou účinností, přes 80 %, proto se využívá například v komerčních bezdrátových nabíječkách pro mobilní telefony. Druhou možností je využití rádiových vln. Jako zdroje energie zde slouží okolní rádiové a televizní stanice, Wi-Fi access pointy a mobilní základové stanice. Jedná se o poměrně neefektivní způsob přenosu energie, takže k získání většího množství energie je potřeba poměrně vysokého vyzářeného výkonu, což zase naráží na zdravotní limity pro expozici rádiovými vlnami. Proto se tento způsob přenosu energie využívá spíše pasivním způsobem, bez použití aktivních vysílačů určených pouze pro napájení okolních senzorů.

Tato technologie je omezená pouze na použití v zástavbě, kde se s velkou pravděpodobností budou v okolí nacházet rádiové vysílače. Pokud se přijímač příliš nepohybuje, jedná se o poměrně předvídatelný a stabilní zdroj. S tímto zdrojem nemůžeme příliš počítat ve volné přírodě, kde je nízká úroveň rádiového šumu. [10]

Tabulka 2.5: Měření intenzity RF signálu v různých prostředích [11]

Spektrum	Prostředí	Energetická hustota (mW na m ²)
GSM (935–960 MHz)	zástavba, venkovní prostory	10^{-3} až 10^{-1}
	uvnitř budovy poblíž okna	10^{-2} až 10^{-1}
GSM (1805–1880 MHz)	50 m od základny	$5 \cdot 10^{-3}$ až 5
	200 m od základny	10^{-3} až 0,5
	500 m od základny	$5 \cdot 10^{-4}$ až $5 \cdot 10^{-2}$
Wi-Fi	v rámci okolí vysílače (<8 m)	10^{-3} až $5 \cdot 10^{-2}$
	12 m od vysílače	10^{-4} až $5 \cdot 10^{-4}$

2.2.4 Pohybové a vibrační zdroje

V závislosti na povaze pohybu se zde nejčastěji uplatňuje elektrostatická a elektromagnetická indukce nebo piezoelektrický jev. Ve venkovním prostředí je asi nejpopulárnějším zdrojem tohoto typu větrná elektrárna. Umožňuje nám získat poměrně silný zdroj energie, ideálně jako doplňkový zdroj k solárním panelům. Dokonce se nemusí nutně jednat o rozměrná zařízení. Existují i experimentální prototypy větrných turbín o rozměrech menších než 1 cm a výkonem 1 mW. Výroba tak miniaturního zařízení už ale vyžaduje použití fotolitografie. [12]

Výkon vibračních zdrojů se značně liší v závislosti na konkrétní konstrukci. Jednou z možných aplikací je například nositelná elektronika, kde můžeme zužitkovat pohyby lidského těla. Při experimentech s použitím vibračního mikrogenerátoru byla získána energie s hustotou $4 \mu\text{W}/\text{cm}^3$. V případě montáže mikrogenerátoru na vibrující stroj je možné dosáhnout až $800 \mu\text{W}/\text{cm}^3$ (2 nm dráha pohybu při frekvenci 2,5 kHz). Další zajímavou aplikací je použití piezoelektrických generátorů v kombinaci se zužitkováním došlapu chodidla při chůzi. Experimenty ukázaly, že je takto možné získat výkon 2–250 mW v závislosti na chůzi a hmotnosti osoby. [13, 10]

Kapitola 3

Bezdrátové sítě typu LPWAN

Označení Low-Power Wide Area Network zastřešuje širší skupinu bezdrátových sítí se specifickými vlastnostmi. V kontrastu s konvenčními telekomunikačními sítěmi, jako jsou 3G, 4G nebo 5G sítě, mají nezvykle nízkou přenosovou rychlost. Někdy se proto můžeme setkat také s označením 0G sítě. Nejedná se tedy o sítě, které by bylo možné použít k přenosu obrazového nebo zvukového záznamu, protože by tento přenos trval neúměrně dlouho. Účelem těchto sítí je zprostředkovat obousměrný komunikační kanál pro velmi vysoké počty koncových zařízení. Aby bylo ale možné obsloužit i několik tisíc zařízení, je velikost přenášených zpráv častokrát omezena na pouhé desítky Bajtů a navíc nesmí souhrnný vysílací čas jednoho zařízení překračovat typicky 36 sekund za hodinu (1% obsazení pásma). Takováto síť slouží převážně pro M2M (machine-to-machine) komunikace a otevírá tím nové možnosti nejen pro průmyslovou automatizaci, ale i zemědělství, správu veřejného prostoru nebo poskytování služeb.

Jak už vyplývá z názvu LPWAN, dalším specifikem je důraz na nízkou energetickou náročnost komunikace pro koncové zařízení. Klíčem k dosažení tohoto cíle je nasazení UNB¹ modulací nebo DSSS². Použití těchto dvou technologií umožňuje detekci rádiové signálu velmi blízko hranici šumu. Malý vysílací výkon vysílače je tak kompenzován vysokou vstupní citlivostí přijímače. Nízká energetická náročnost nám tak umožňuje vybavit rádiovým spojením i zařízení, která mají omezené zdroje energie a zvýšit tak jejich užitnou hodnotu. Dále se zde rýsuje zcela nová oblast různých typů bezdrátových senzorů, které jsou schopny provozu po dobu několika let, a zároveň nevyžadují po dobu svojí životnosti žádnou údržbu v podobě výměny baterií. Dalším možným využitím těchto sítí je geolokace koncových zařízení. S využitím stacionární sítě pozemních vysílačů je možné určit na základě síly signálu jednotlivých vysílačů přibližnou polohu koncového zařízení. Přesnost takto vypočtené polohy je prozatím okolo 1 až 10 km. V budoucnu bude ale s hustší infrastrukturou vysílačů možné určovat polohu s chybou okolo 100 metrů. [14]

¹Ultra narrow band

²Direct sequence spread spektrum – Technika rozprostřeného spektra

V současné době se o vývoj hardwaru a různých typů sítí zajímá hned několik společností, které častokrát zakládají sdružení a budují si zákaznické komunity. Díky tomu je v současnosti na trhu hned několik technologií. Ze zákaznického pohledu je toto výhodná situace, protože si můžeme zvolit řešení přesně na míru našim požadavkům. Každá z technologií má navíc jinak nastavený platební model, což nám dává manévrovací prostor pro stanovení optimální technologie. Konkrétní detaily jednotlivých technologií budou podrobněji rozebrány v následujících kapitolách.

3.1 LoRaWAN

LoRaWAN je otevřený standard LPWAN určený pro IoT zařízení. O jeho specifikaci a další vývoj se stará LoRa Alliance, což je otevřené neziskové sdružení zaměřené na jeho vývoj a propagaci. Aliance sdružuje společnosti jako je Cisco, HP, Semtech, Schneider atd. Nedílnou součástí jsou mobilní operátoři, kteří nasazují síťové prvky do svojí stávající infrastruktury. Navzdory tomu, že se jedná o otevřený standard, je prozatím fakticky jediným dodavatelem hardwaru společnost Semtech. To by se ale do budoucna mohlo změnit. Společnosti Microchip, ST Microelectronics a Renesas, projevíli zájem o integraci LoRa technologie do svých zařízení, takže můžeme v blízké době očekávat spuštění výroby i u dalších výrobců integrovaných obvodů.

LoRa Alliance provozuje síť soukromých laboratoří za účelem certifikace produktů třetích stran. Výrobce hardwaru může požádat o certifikaci vlastního hardwaru podle specifikace LoRaWAN a následně obdržet ochrannou známku LoRaWAN® Certified^{CM}. Pro členy aliance je certifikace zcela zdarma. [15]

Specifikace LoRaWAN definuje první tři síťové vrstvy ISO-OSI modelu. Na fyzické vrstvě je použita modulace LoRa, která je kombinací CSS³ techniky a DSSS⁴ modulace. Podrobnosti linkové a síťové vrstvy jsou pak uvedeny ve specifikaci LoRaWAN. Implementace vyšších vrstev je už plně v rukou konkrétního aplikačního serveru.

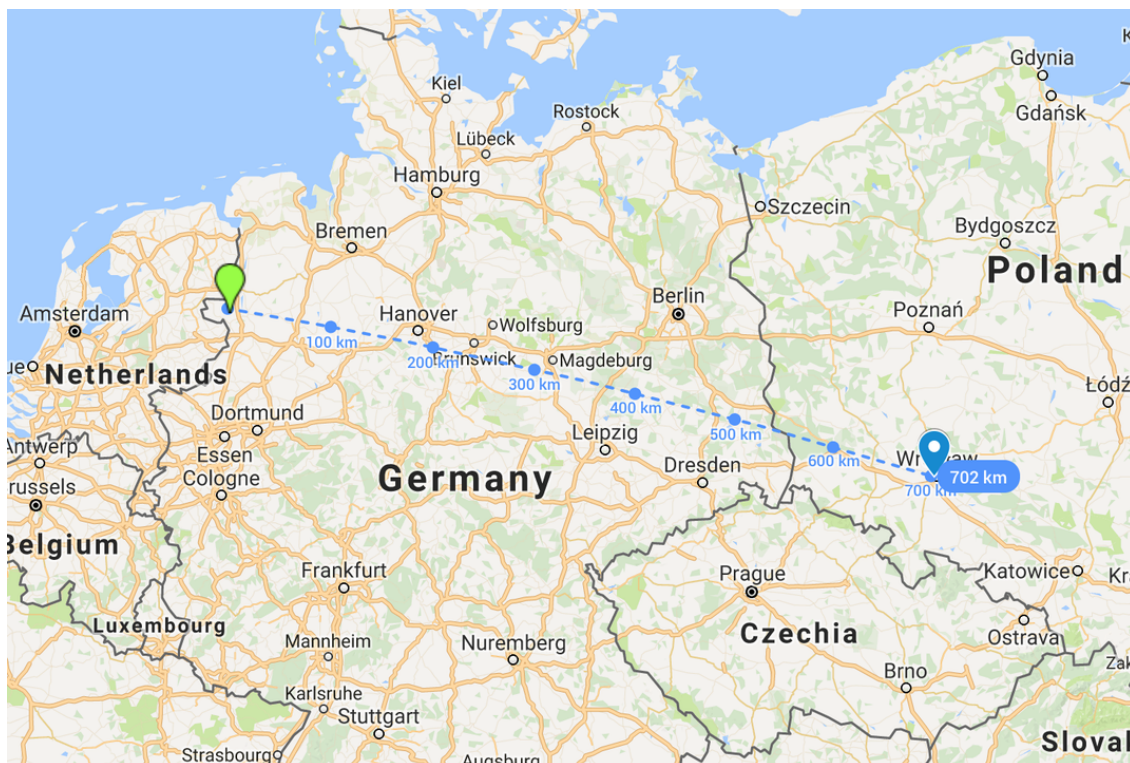
Technologie LoRaWAN nám nabízí možnost obousměrného přenášení zpráv na vzdálenost až 15 km ve volném prostoru nebo 5 km v zástavbě. Při přímé viditelnosti mezi vysílačem a bránou je dokonce možné dosáhnout i o jeden řád větší přenosové vzdálenosti. Byl spolehlivě zaznamenán přenos z heliového balónu na vzdálenost 702 km. Vysílací výkon byl přitom pouhých 25 mW. [16, 17]

3.1.1 Fyzická vrstva – LoRa

Vysílání na tak velké vzdálenosti, při současném dodržení max. vyzářeného výkonu 25 mW, vyžaduje nasazení velmi citlivých přijímačů. V případě LoRa brány je možné dosáhnout vstupní citlivosti přijímače až -137 dBm. Přijímaný signál se častokrát nachází až pod hranicí rádiového šumu, takže

³Chirp spread spectrum

⁴Direct sequence spread spectrum



Obrázek 3.1: Rekordní přenos na vzdálenost 702 km [16]

je v tomto ohledu velmi těžce vystopovatelný. Vysílání probíhá v pásmech 433 MHz, 868 MHz nebo 915 MHz.

Základem LoRa modulace je nosná vlna v podobě chirp signálu, který v průběhu vysílání plynule mění svou frekvenci. Rozsah změn frekvence je dán přednastavenou šířkou pásma. Pro uplink se používá šířka pásma 125 kHz nebo 500 kHz a 500 kHz pro downlink. Tím, že dochází k plynulé změně nosného kmitočtu, je signál odolný proti Dopplerovskému efektu (jeho vliv je zanedbatelný v poměru s rychlostí rozmítání signálu), takže se tato modulace hodí i pro pohyblivé aplikace. Kompenzaci drobných odchylek frekvence nosné vlny jsme schopni realizovat na straně přijímače. Konstrukce vysílače tak nevyžaduje zdroj přesného hodinového signálu, což má za následek další zlevnění vysílacích modulů.

K dosažení ještě větší citlivosti přijímače jsou vysílané bity zakódované technikou, která byla převzata a upravena ze systému DSSS. Každý datový bit je nahrazen určitou sekvencí symbolů. Délka této sekvence je v terminologii LoRa označována jako SF⁵. Čím je tato sekvence delší, tím větší citlivosti jsem schopni dosáhnout na straně přijímače a získáváme větší dosah signálu. Zvýšení citlivosti je způsobeno tím, že zvyšujeme redundanci informací ve vysílaném signálu. Delší sekvence symbolů také znamená nižší komunikační rychlost. Faktor šíření může být dynamicky měněn podle aktuální kvality signálu. Pokud se koncové zařízení nachází v blízkém okolí brány, může si díky dobré

⁵SF – Spreading Factor – faktor šíření

kvalitě signálu dovolit vyšší SF, tzn. dojde ke zvýšení přenosové rychlosti a zkrácení doby vysílání, což se projeví snížením spotřeby koncového zařízení. Modulace LoRa má celkem šest faktorů šíření (označovaných jako SF7 až SF12). [17]

Faktor šíření	Bitová rychlost	Orientační dosah	Doba vysílání paketu
SF10	980 bps	8 km	371 ms
SF9	1 760 bps	6 km	185 ms
SF8	3 125 bps	4 km	103 ms
SF7	5 470 bps	2 km	61 ms

Tabulka 3.1: Možné faktory šíření pro šířku pásma 125 kHz [17]

Faktor šíření SF11 a SF12 je možné použít pouze pro šířku pásma 500 kHz.

3.1.2 Řešení kolizí na fyzické vrstvě

Při vysokých počtech koncových zařízení s tak velkým vysílacím dosahem vzrůstá pravděpodobnost vzniku kolize. Vzhledem k tomu, že se do budoucna počítá s masivním využitím tohoto typu komunikace, nemůžeme spoléhat pouze na 1 % omezení využití vysílacího času a náhodné vysílací intervaly. Proto je součástí modulace ještě jeden mechanismus omezující počet vzniklých kolizí. Každý z tzv. faktorů šíření se odlišuje nejen délkou symbolické sekvence představující jeden datový bit, ale i vzorem pro jednotlivé bity. Jednotlivé vzory jsou vůči sobě navzájem ortogonální. Pokud dojde shodou okolností k odeslání dvou různých paketů ve stejný čas, ale s rozdílným faktorem šíření, bude brána stále schopna demodulovat současně oba pakety. Pouze v případě současného vysílání dvou zařízení se stejným faktorem šíření dojde k interferenci signálů, ale i v tomto případě je možné přijmout alespoň jeden paket, pokud bude rozdíl sil signálu paketů větší než 6 dB. Při dostatečně hustém pokrytí oblasti několika bránami LoRaWAN se ale ani tak slabší paket pravděpodobně neztratí. Častokrát je jeden stejný paket zachycen hned několika bránami zároveň, takže případná interference signálu na některé z bran neznámá automaticky ztrátu dat. [17]

3.1.3 Zabezpečení komunikace

Síť LoRaWAN implementuje několik bezpečnostních mechanismů pro zajištění bezpečné komunikace mezi aplikací a koncovým zařízením. Každé připojené zařízení v síti se musí autorizovat, aby s ním síť zahájila komunikaci. Autorizaci zařízení je možné provést buď metodou OTAA⁶, nebo metodou ABP⁷. Dalším bezpečnostním prvkem je ochrana integrity přenášených zpráv a dvojí šifrování pomocí 128bitové AES šifry. Koncové zařízení nejprve pomocí aplikačního klíče AppSKey

⁶Over the air activation

⁷Activation by personalization

zašifruje přenášená data. Klíč je znám pouze aplikačnímu serveru a koncovému zařízení. K paketu jsou následně přidány identifikátory zařízení a další informace potřebné pro komunikaci se sítí LoRaWAN. Sestavený paket je znovu zašifrován tentokrát pomocí tzv. NwkSKey – network session key (síťového relačního klíče). Tento klíč už je znám všem serverům obsluhujícím bránu sítě LoRaWAN, takže mohou rozšifrovat hlavičku zprávy a zajistit doručení dat příslušnému aplikačnímu serveru. Obsah datové části paketu jim ale ani tak není znám. [17]

Pokud by se útočník pokusil o zachycení rádiového přenosu, nejprve by se musel potýkat s tím, jak vůbec signál zachytit, když je přijímaný signál častokrát pod hranicí přirozeného rádiového šumu. Díky šifrování by ale stejně nezjistil žádné identifikátory vysílače. Opakované odvysílání zachyceného paketu by také k ničemu nevedlo, protože součástí hlavičky je pořadové číslo paketu, které umožňuje síťovým serverům odlišit duplicitní pakety, které mohou v síti vznikat i přirozeným způsobem, když je paket přijat více bránami. Automaticky tak dochází k zahazování paketů, které nemají správné pořadové číslo zprávy pro dané zařízení. V nastavení aplikačního serveru je možné tuto filtraci paketů deaktivovat, nicméně se tato možnost v žádném případě nedoporučuje pro nasazení na produkční aplikace. Je zde jen pro usnadnění testování nových zařízení.

Komunikace mezi aplikací a koncovým zařízením je tedy chráněna proti:

- Pozměnění obsahu paketu.
- Připojení cizího zařízení do sítě.
- Odposlechu přenášených dat.
- Odchycení a opakování zachyceného paketu.

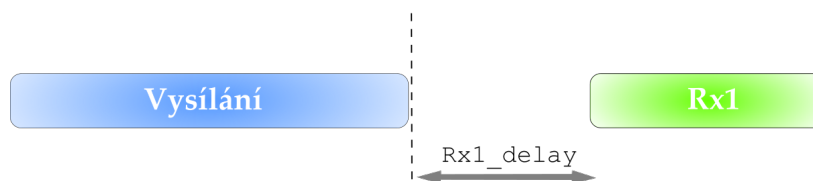
3.1.4 Komunikační třídy

Koncová zařízení mohou komunikovat ve 3 různých režimech – komunikačních třídách. Každá třída se hodí pro jiný druh aplikace, podle toho jestli upřednostňujeme malou spotřebu energie nebo rychlost odezvy a větší objem přenášených dat. Pro všechna zařízení platí, že musí minimálně podporovat třídu A. Zařízení pro třídu B musí podporovat i třídu A. Podobně tak zařízení ve třídě C musí umět komunikovat ve třídě A i B.

3.1.4.1 Třída A

Komunikace je v této třídě inicializována koncovým zařízením. Typickým kandidátem pro komunikaci v této třídě jsou různé typy bezpečnostních detektorů (kouře, kapalin, nebezpečných plynů) napájených z baterií. Zařízení se nachází po většinu času v režimu spánku, aby ušetřilo energii. Neprobíhá žádná komunikace se sítí. V okamžiku výskytu alarmu se aktivuje rádiový modem a zahájí vysílání informací o sledovaném ději. Po dokončení přenosu zařízení přejde opět do režimu spánku

na dobu jedné vteřiny (interval je nastavitelný). Po uplynutí tohoto intervalu se zařízení znovu aktivuje a přijme odpověď od nejbližší brány.



Obrázek 3.2: Aktivita zařízení při úspěšném příjmu v oknu Rx1

Pokud není detekována žádná aktivita na přijímacím kanále, zařízení nečeká na uplynutí celého přijímacího okna, ale předčasně přejde do režimu spánku a vyčkává do druhého přijímacího okna.



Obrázek 3.3: Aktivita zařízení při úspěšném příjmu v oknu Rx2

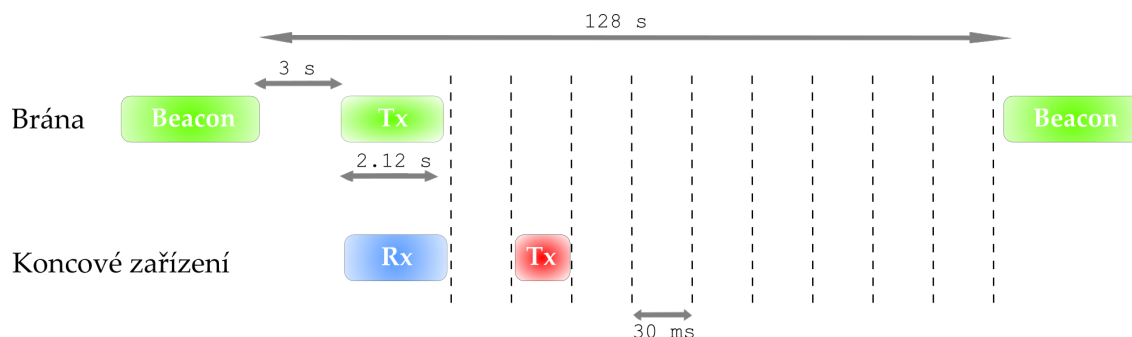
Jestliže ani v druhém přijímacím okně není zachycena žádná odpověď brány, nebyl vysílaný paket pravděpodobně zachycen žádnou z okolních bran a je potřeba opakovat vysílání. Vysílání musí být ale v souladu s pravidly pro maximální povolené využití pásma v čase.

Pro tuto třídu neexistuje způsob, jak by mohlo být koncové zařízení kontaktováno serverem bez toho, že samo inicializuje komunikaci a následně může přijmout nějaké řídicí povely.

3.1.4.2 Třída B

Tato třída je určena pro všechna zařízení, která potřebují v pravidelných intervalech odesílat data, nebo v aplikacích, kde je nutné pravidelně zasílat koncovým zařízením řídicí povely. K zajištění synchronizace komunikace vysílají brány každých 128 sekund synchronizační signál. Synchronizační signál obsahuje informaci o aktuálním času a kontrolní součet + 7 Bajtů dalších informací (informace záleží na typu brány). Krátce po synchronizačním signálu, označovaném jako beacon, pošle brána data koncovým zařízením a následně čeká na jejich odpovědi.

Protože může být jedna zpráva směřována hned několika koncovým zařízením zároveň, musí koncová zařízení odesílat odpověď s různým zpožděním, aby nedocházelo zbytečně ke kolizím. Celkem je k dispozici 4096 časových slotů o délce 30 ms, ve kterých mohou koncová zařízení odeslat odpověď bráně.

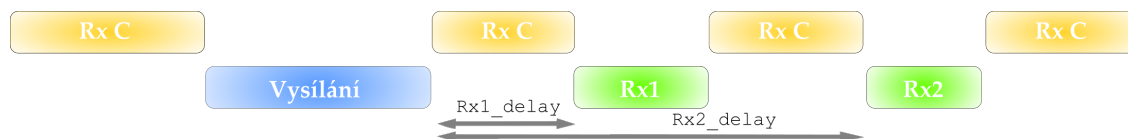


Obrázek 3.4: Průběh komunikace v třídě B

3.1.4.3 Třída C

Pro zařízení se stálými zdroji energie, jako jsou pouliční lampy, elektroměry nebo závory, byla vytvořena tato komunikační třída. Komunikace z brány ke koncovému zařízení může proběhnout téměř kdykoliv, zařízení je vždy připraveno zahájit příjem nových dat. Výjimkou je krátký interval, kdy zařízení odesílá data bráně. Tento komunikační režim má ze všech výše uvedených tříd nejmenší komunikační zpoždění, ale také nejvyšší energetickou náročnost na provoz.

Na časovém průběhu komunikace můžeme vidět, že koncové zařízení může přijímat data hned poté, co dokončí vysílání. Stále ale platí, že po vysílání koncového zařízení existují dvě časová okna pro potvrzení zprávy. Pokud by v tomto intervalu brána vysílala data, musí tento přenos přerušit a odvysílat odpověď na přijatou zprávu. Následně může pokračovat ve vysílání přerušené relace. I u této třídy komunikace se doporučuje používat počítadla odchozích a příchozích paketů, aby nemohlo dojít k narušení komunikace útočníkem, který by mohl provést zachycení a zopakování paketu.



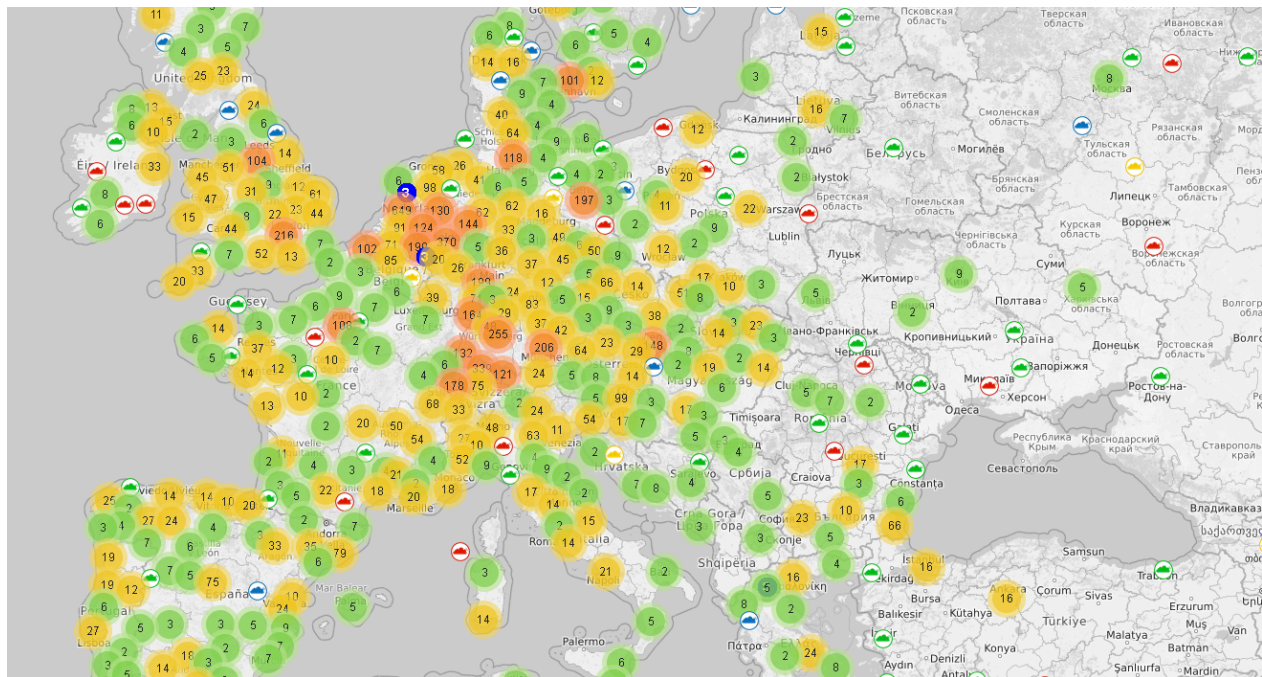
Obrázek 3.5: Průběh komunikace v třídě C

Zařízení nemusí komunikovat celou dobu ve třídě C. Pokud vybavíme koncové zařízení potřebným softwarem, můžeme jej na dálku přepnout do jiné komunikační třídy a snížit tak dočasně spotřebu energie. Pomocí přepínání komunikačních tříd můžeme dosáhnout celkem zajímavých parametrů komunikace. Můžeme například provozovat síť senzorů s aktuátory v komunikační třídě B. Koncová zařízení tak mohou mít velmi nízkou spotřebu. V okamžiku, kdy senzory zaznamenají nějakou anomálii, může aplikační server poslat určitým zařízením příkaz k přepnutí komunikace do třídy C, čímž zkrátíme komunikační odezvu a můžeme začít provádět na dálku kroky k zabránění dalších škod.

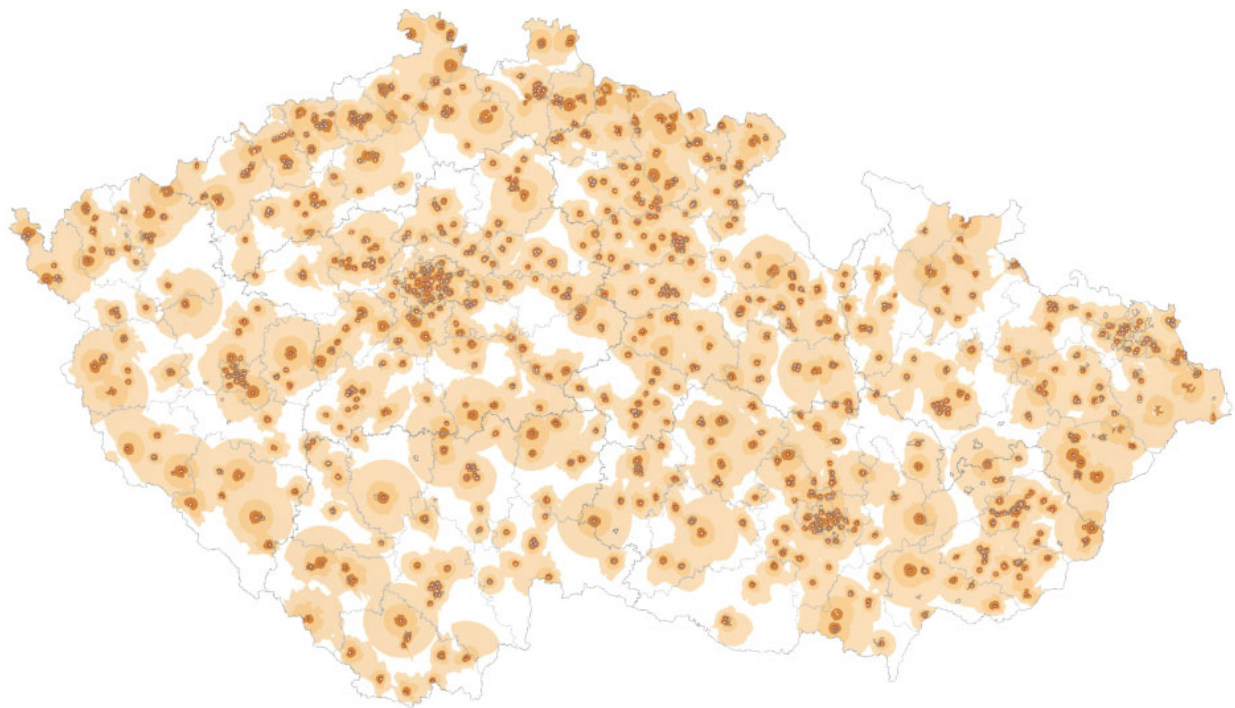
3.1.5 Infrastruktura a pokrytí

Stanovení pokrytí sítě LoRaWAN není úplně jednoznačná záležitost a ne vždy lze s úplnou přesností zjistit, zda je daná lokalita v dosahu sítě. Neexistuje totiž centrální mapa, která by zachycovala síťovou infrastrukturu všech zúčastněných subjektů. Sít LoRaWAN je totiž zčásti budována komerčními společnostmi a částečně komunitou. Navíc existuje hned několik druhů backendových serverů a několik provozovatelů této služby. Nelze tedy jednoduše shromáždit informace o všech instalovaných síťových branách. Obecně můžeme prohlásit, že asi nejhustěji pokrytou oblastí k roku 2021 je Evropa s výjimkou východní části. Zde můžeme mluvit o téměř plošném pokrytí celého území. Velký počet přístupových bran se nachází i v USA. Zdaleka se už ale nejedná o plošné pokrytí, ale spíše jen o pokrytí velkých aglomerací.

Výhodou sítě LoRaWAN je její komunitní zaměření. Kdokoliv si může opatřit potřebné díly a pokrýt signálem nejbližší okolí. Navíc existence hned několika provozovatelů backendových serverů pro provoz sítě nám dává určitou záruku do budoucna. V případě ukončení provozu některého z provozovatelů je možné napojit stávající infrastrukturu na jiného poskytovatele.



Obrázek 3.6: LoRaWAN síť v režiji TheThingsNetwork 3/2021 [18]



Obrázek 3.7: LoRaWAN síť v režiji Českých radiokomunikací 6/2020 [19]

3.1.6 Poplatky

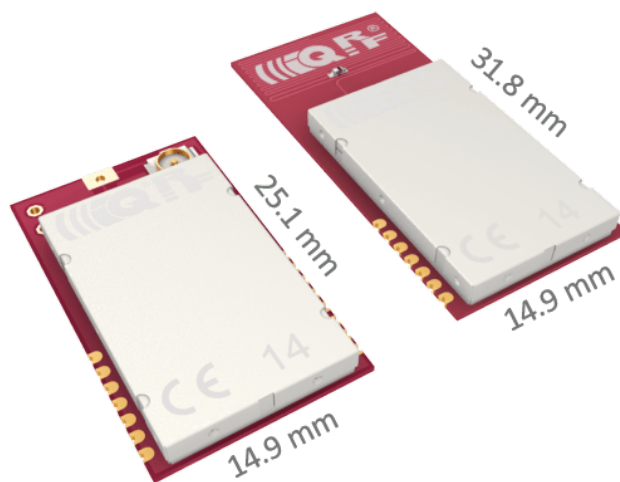
Síť LoRaWAN pracuje v bezlicenčním pásmu, proto nemusíme platit žádné licenční poplatky za využití rádiového pásma. Jediné náklady plynou z pořízení a provozu zařízení. Pokud se rozhodneme připojit naše zařízení do sítě, která je volně dostupná komunitě, budou naše jediné náklady spojené s pořízením zařízení.

Druhou možností je oslovit některý komerční subjekt, který provozují vlastní backend servery a síťové brány. Zde se cenová kalkulace odvíjí od počtu zařízení. Typická cena je 200–400 Kč za připojení jednoho zařízení na jeden rok. Častokrát poskytovatel nabízí rovnou hotová zařízení k pronájmu i se zajištěním připojení do sítě.

3.2 IQRF

IQRF je univerzální bezdrátová platforma pro vytváření mesh sítí. Na rozdíl od ostatních technologií nevyžaduje žádnou síťovou infrastrukturu třetích stran, vše se buduje v místě nasazení na míru. Jedná se o paketově orientované řešení založené na nízkém vysílacím výkonu v kombinaci s přeposíláním paketů napříč celou sítí. Technologie IQRF není jen bezdrátovou sítí, ale zároveň celým spektrem hardwaru a softwarových nástrojů určených pro snadnou a rychlou integraci této

technologie do různých aplikací. Jedná se o skutečně flexibilní řešení, které nabízí nejen připravená typová řešení pro rychlé nasazení, ale i poměrně nízkourovňový přístup k rádiovým modulům, pokud to daná aplikace vyžaduje.



Obrázek 3.8: Rádiový modul IQRF

Základem celé technologie jsou malé a vysoce kompaktní rádiové moduly vybavené mikrokontrolérem. Mikrokontrolér už obsahuje tyto periferie a komponenty (existují zde drobné rozdíly v závislosti na modelu):

- Až 12 GPIO pinů.
- SPI, I²C, UART a další rozhraní.
- A/D a D/A převodníky.
- Analogový komparátor.
- PWM výstup.
- EEPROM paměť.
- Teplotní senzor.
- LED indikátory.
- Stabilizátor napětí.

Díky tomu už není potřeba přidávat k zařízení příliš mnoho dalších komponent, ale je možné rovnou využít možností integrovaného hardwaru. [20]

3.2.1 Parametry rádiového vysílače

Rádiové vysílání nemá jednotnou modulaci signálu. V závislosti na typu vysílače a verzi operačního systému zařízení se můžeme setkat s ASK⁸, FSK⁹ nebo GFSK¹⁰ modulací. Tyto 3 druhy nejsou navzájem kompatibilní, není možné je spojit do jedné sítě. Vysílací výkon se odvíjí podle toho, v jakém pásmu právě pracuje daná síť. Pro pásma 868 a 916 MHz je nejvyšší vysílací výkon 3.5 mW, pro pásmo 433 MHz až 5 mW. Při výběru pásma je potřeba respektovat místní pravidla rádiového provozu. Dosah jednoho vysílače je uvnitř budov několik desítek metrů, pro venkovní prostředí až několik stovek metrů. Pokud ale k modulu připojíme externí směrovou anténu, je možné takto vytvořit spojení až na 1 kilometr.

Vysílání je rozděleno podle tzv. kanálových map, které rovnoměrně rozdělují celé frekvenční spektrum po 100 kHz na jednotlivé komunikační kanály. Každý základní kmitočet má jiný počet pásem. Pro pásmo 433 MHz existuje celkem 16 kanálů v rozmezí 433,10–434,70 MHz. Pásmo 868 MHz je rozděleno na celkem 67 kanálů v rozmezí 863.15–869.85 MHz. Ne všechny kanály je ale dovoleno použít, protože jsou rezervovány jinou službou. Například v prostoru EU není dovoleno použít kanály 61–67, protože jsou vyhrazeny pro poplachové a zabezpečovací systémy. Navíc se na některé kmitočty vztahují další omezení ohledně maximálního hodinového obsazení rádiového pásma. Při volbě pracovního kanálu je proto potřeba vzít v úvahu i tyto faktory. Pásmo 916 MHz je rozděleno na celkem 255 kanálů v rozpětí 902,25–927,75 MHz. Existuje zde ale jedno výrazné omezení pro území Izraele, kde je dovoleno použít pouze kanály 133–140, tj. 915,55–916,25 MHz. [21]

3.2.2 Komunikace v rámci sítě

IQRF podporuje dva diametrálně odlišné způsoby komunikace. Výchozím způsobem komunikace je **režim peer-to-peer**. V tomto režimu je možné komunikovat s libovolným počtem zařízení. Všechna zařízení přijímají veškerou komunikaci v síti. Velikost sítě je omezená na dosah jednoho vysílače, protože je potřeba, aby mezi sebou měla všechna zařízení přímé rádiové spojení. V praxi to znamená při dobrých podmínkách šíření signálu na vzdálenost okolo 500 m. Průběh komunikace je plně dán uživatelským programem. Tento režim je vhodný pro aplikace, kde počítáme s pohybem koncových zařízení, takže se nám plynule mění vzdálenosti mezi zařízeními.

Skutečná síla této technologie spočívá v režimu komunikace s názvem **IQMESH**. Síť se v tomto režimu skládá z jednoho koordinátora, řídícího celý provoz sítě a až 239 uzlových zařízení. Klíčovou vlastností tohoto režimu je schopnost uzlů předávat si navzájem pakety ze vzdálených zařízení a překlenout tak větší vzdálenosti. Napříč celou sítí je možné obousměrně posílat mezi uzly až 64 Bajtů dat v jednom paketu. Pro směrování paketů v síti se používá patentovaný způsob předávání

⁸Amplitude shift keying

⁹Frequency shift keying

¹⁰Gaussian frequency shift keying

paketů označovaný jako directional flooding – směrové zaplavování. Tento způsob komunikace se řídí následujícími principy:

- Každý uzel vysílá paket pouze jednou.
- Vysílání může probíhat pouze ve vyhrazeném časovém slotu. Aby se zabránilo kolizím při vysílání, je používán časový multiplex typu TDMA (Time division multiple access).
- Koordinátor ani příjemce neprovádí další směrování paketu.
- Uzly se nesmí pohybovat – jakákoliv změna polohy uzlů vyžaduje opětovné spuštění průzkumu sítě a určení možných signálových cest v síti. Tento proces je ale poměrně energeticky náročný. V závislosti na velikosti sítě může trvat od několika desítek sekund po několik minut. Po tuto dobu není síť schopná přenášet jakákoliv data.
- Komunikace probíhá synchronně a je zahajována koordinátorem. Asynchronní komunikace pro výměnu dat mezi uzly navzájem je sice možná, ale aplikační software musí zajistit, aby nevznikaly kolize se standardním provozem sítě.

Primárně probíhá komunikace vždy mezi koordinátorem sítě a jedním vybraným zařízením v síti. Zasláný paket je postupně přeposílán všem sousedním uzlům až nakonec dorazí k adresátovi. Stejným způsobem putuje paket s odpovědí zpět ke koordinátorovi. Odvysílání jednoho paketu při přenosové rychlosti 19,836 kb/s trvá v závislosti na objemu přenášených dat okolo 30–50 ms. S rostoucím počtem uzlů se zvyšuje počet potřebných skoků paketu, než dojde k doručení zprávy do cíle. Extrémním případem je stav, kdy potřebujeme získat data ze všech uzlů v síti nebo provést hromadnou akci na všech připojených uzlech. Postupné vyčítání dat není z praktického hlediska příliš vhodné. Z důvodu exponenciální závislosti času odezvy na počtu uzlů v síti se už při relativně malém počtu uzlů dostaneme na časy odezvy v řádu desítek vteřin. Z těchto důvodů byl do komunikačního protokolu implementován mechanismus pro provádění hromadných operací.

- Fast response command – FRC

Pro účely provádění hromadných akcí všech uzlů s možností získání zpětné vazby existuje právě FRC paket. Tento paket může inicializovat pouze koordinátor. Každý uzel může navíc poslat zpátky krátkou odpověď na tento paket. Velikost zpětného paketu nemůže přesáhnout 64 Bajtů, takže je velikost zpětné zprávy omezená podle toho, kolika uzlů se povel týkal. V případě zavolání všech 239 možných uzlů je velikost zpětného paketu pouze 2 bity, do 63 uzlů 1 Bajt, do 15 uzlů 4 Bajty, atd. Tento typ paketů se například hodí pro hromadné zapínání spotřebičů.

Odbavení FRC paketu můžeme z hlediska uzlu rozdělit do 3 kroků. V první fázi dochází k šíření tohoto paketu napříč celou sítí ke všem uzlům. Následně provedou všechny uzly požadovanou operaci, připraví si data pro odpověď koordinátorovi sítě a rozešlou tuto odpověď

sousedním uzlům. Posledním krokem je zahájení přenosu odpovědi zpět ke koordinátorovi. Tento paket s odpovědí inicializuje nejvzdálenější uzel sítě. Protože poslední uzel již má informace o výsledku operace u sousedních uzlů, obsahuje nově sestavený odpovědní paket data jak z posledního uzlu, tak i okolních uzlů. Paket s odpovědí putuje postupně zpátky směrem ke koordinátorovi. Každý uzel, který jej zachytí, přidá do jeho obsahu svou odpověď a odvysílá jej všem okolním uzlům. Postupně tak dochází ke sběru informací z celé sítě. Nakonec může dorazit ke koordinátorovi sítě hned několik replik paketů, které obsahují odpovědi uzlů na trase, kterou daný paket cestoval. Část přijatých informací může být duplicitní, ale máme tak jistotu, že získáme data ze všech uzlů v síti. V závislosti na tom, v jakém režimu pracují síťové uzly, můžeme u sítě s 239 uzly získat kompletní odpověď v rozmezí 33,9–54,4 sekund (standardní režim) nebo v rozmezí 46,0–66,5 sekund (low-power režim).

- **Selektivní FRC**

Umožňuje provádět skupinové akce s vybranými uzly.

- **Lokální FRC**

Umožňuje asynchronní výměnu povelů mezi uzly. Paket má omezené působení pouze na rádiový dosah zdrojového vysílače, paket se nepřeposílá dál po síti. Jeho výhodou je téměř nulové zpoždění, takže se například výborně hodí pro ovládání akčních členů v rámci jedné místnosti.

[20]

Počet uzlů v síti	Doba vyčtení dat postupným dotazováním [s]	Doba vyčtení dat použitím FRC [s]
10	3	1
20	13	2
50	77	6
100	303	12
200	1206	24
239	1721	34

Tabulka 3.2: Přibližná doba potřebná ke sběru dat ze všech uzlů v síti

3.2.2.1 Určení struktury sítě

IQRF má zabudovaný algoritmus pro automatickou analýzu struktury mesh sítě. Prvním krokem pro vytvoření sítě je spárování všech zařízení (uzlů) s daným koordinátorem, tzv. bonding. Existuje více postupů, jak lze provést toto spárování, ale nechci zde zbytečně zabíhat do detailů. Podstatou spárování je nastavení přístupových klíčů a identifikátorů uzlů.

Když jsou všechna IQRF zařízení na svých místech, můžeme zahájit proces mapování sítě. Mapování sítě je plně v režii koordinátora sítě. Existují celkem 3 možné strategie mapování sítě:

- DFM – Discovered full mesh
Koordinátor zjistí počet dostupných uzlů ve svém dosahu a přiřadí jim tzv. VRN – Virtual routing number. Následně postupně každý detekovaný uzel detekuje další uzly ve svém dosahu a přiřazuje jim vyšší VRN, než je jeho vlastní VRN. Informaci o nově přiřazených VRN posílá uzel zpátky koordinátorovi. S rostoucí vzdáleností uzlu od koordinátora se tak hodnoty VRN zvyšují, zároveň v síti neexistují 2 zařízení se stejným VRN číslem. VRN následně slouží k adresaci paketů v síti a odvíjí se od nich časová synchronizace komunikace.
- DOM – Discovered optimized mesh
- DRM – Discovered reduced mesh
V obou případech se provádí redukování počtů přeskoků paketů mezi uzly. Snižuje se tak doba potřebná k přenosu dat, ale i počet redundantních spojení v síti, a tím klesá robustnost sítě.[22]

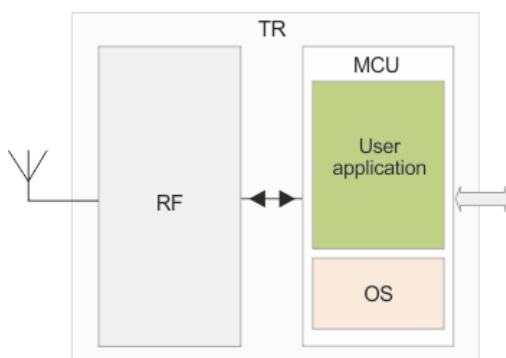
3.2.3 Software a vývoj

Všechna zařízení IQRF obsahují speciální operační systém. Operační systém je průběžně vylepšován, takže existuje v mnoha verzích. Výrobce dodává hardware s nejnovější verzí OS, na vyžádání je možné opatřit danou objednávku i libovolnou starší verzí OS. V určitých rozmezích verzí je ale možné kdykoliv přeprogramovat OS na jinou verzi. Důležité je, aby všechna zařízení v jedné síti používala stejnou verzi OS. Pokud provádíme výměnu IQRF zařízení za novější typ, je potřeba si zkontrolovat kompatibilitu nového zařízení s verzí OS, který používá daná síť. Existují zde totiž jistá omezení, co lze a nelze na jednotlivých typech hardwaru provozovat z hlediska verzí OS. Podrobnosti je možné zjistit na stránkách výrobce.

Operační systém zajišťuje základní operace s hardwarem. Najdeme zde funkce pro:

- Správu režimů spánku.
- Časovací funkce.
- Ovládání LED indikátorů.
- Práci s interní a externí EEPROM.
- Operace s datovými buffery.
- Nastavování rádiové sítě.
- Šifrovací funkce.

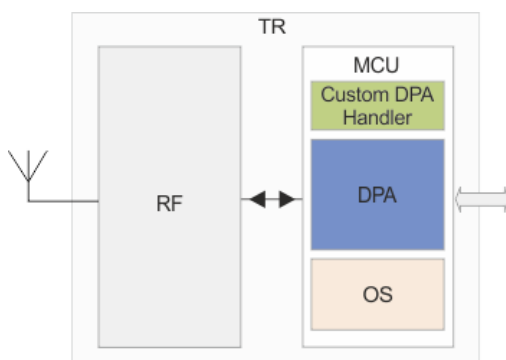
Pro jednoduché aplikace, bez použití mesh sítě, je možné zařízení provozovat v tzv. dvouúrovňové architektuře. Do zařízení se nahraje pouze OS a uživatelský program. Uživatelský program se vytváří v jazyce C.



Obrázek 3.9: Struktura aplikace bez použití DPA [23]

Použití mesh sítě vyžaduje 3úrovňovou architekturu. Kromě operačního systému se musí do zařízení nahrát DPA¹¹ handler. Jedná se o hotový software připravený výrobcem. Tyto knihovny obsahují všechny software, který je potřebný pro provoz mesh sítě. Programátor se tam nemusí zabývat problematikou provozu rádiové sítě s mesh topologií a může se soustředit na vlastní aplikaci.

Vlastní aplikace (označovaná jako Custom DPA Handler) se opět vytváří v jazyce C. Pokud bychom se ale chtěli za každou cenu vyhnout programování vlastního aplikačního softwaru, je možné využít hotových ukázkových programů. Na webové stránce¹² je možné si stáhnout hotové programy s nejčastěji řešenými úlohami v automatizaci (generování PWM, komunikace pomocí sériových linek, měření teploty atd.).



Obrázek 3.10: Struktura aplikace s DPA [24]

Pro práci s IQRf zařízeními nabízí výrobce zdarma komplexní vývojové prostředí. Vývojové prostředí existuje ve dvou variantách. Varianta s grafickým rozhraním umožňuje interaktivní práci

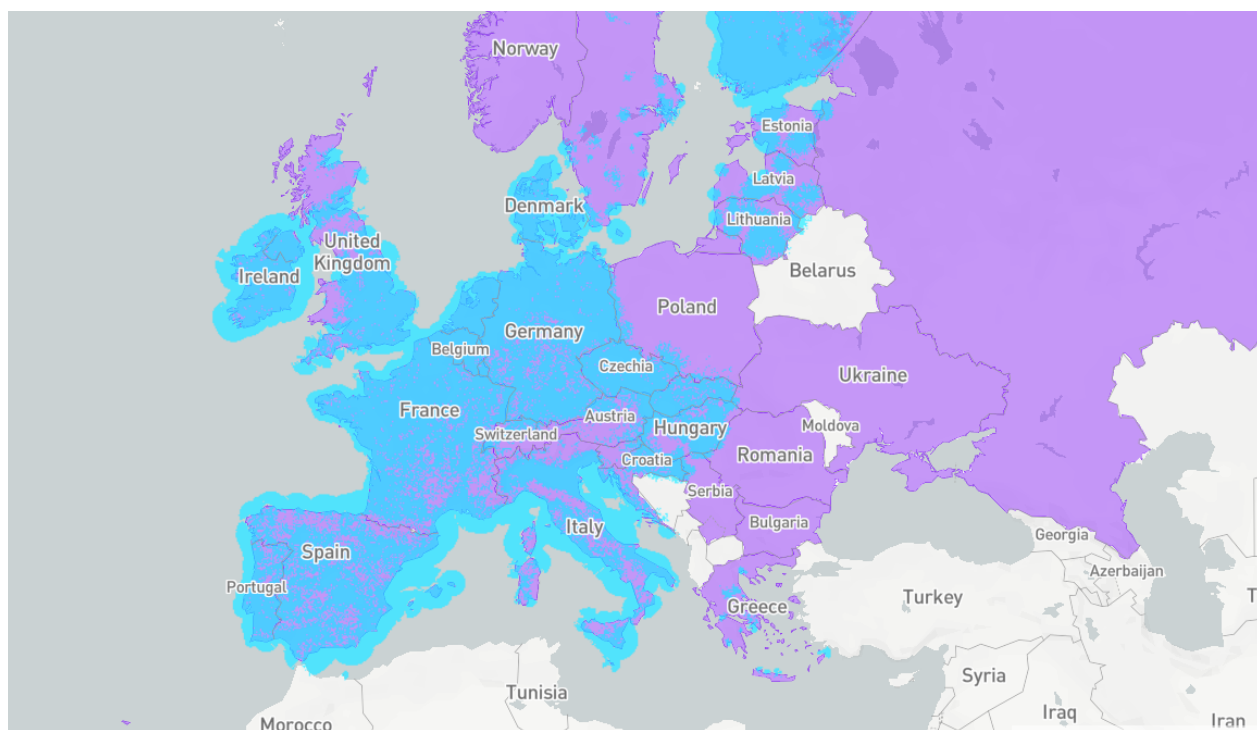
¹¹Direct peripheral access

¹²<https://doc.iqrf.org/DpaTechGuide/examples/>

se zařízeními, ladění softwaru, analýzu rádiového provozu, upgrade OS rádiových modulů a mnoho dalších věcí. Druhou variantou je vývojové prostředí s příkazovou řádkou. Toto je určeno spíše pokročilejším uživatelům. Umožňuje automatizovat rutinní úkony pomocí připravených skriptů.

3.3 Sigfox

Síť Sigfox má svůj počátek ve Francii. Za jejím vznikem v roce 2010 stojí stejnojmenná společnost, která spolupracuje s četnými partnery, jako jsou Texas Instrument, Silicon Labs nebo ON Semiconductor. Cílem této technologie je vytvoření vysokokapacitní nízkoenergetické sítě s dlouhým dosahem. V současné době je tato společnost ve fázi budování a rozšiřování síťové infrastruktury prostřednictvím mobilních operátorů. Společnost investuje vysoké částky do globální expanze svého ekosystému formou spolupráce s lokálními mobilními operátory. V případě České republiky se jedná o společnosti T-Mobile a SimpleCell. [25]



Obrázek 3.11: Pokrytí Evropy sítí Sigfox v roce 2021 (modře), ve výstavbě (fialově) [26]

3.3.1 Pokrytí

Samotná společnost Sigfox neprovádí budování svojí sítě vysílačů, ale přenechává budování infrastruktury na svých partnerech, lokálních mobilních operátorech. Sigfox zajišťuje pouze jednotnou cloudovou infrastrukturu pro všechny smluvní partnery. Díky tomu má tato síť celkem efektivně

vyřešený roaming služby napříč různými státy. V současné době je pokryto přes 72 států s celkovým počtem 17,2 milionů zaregistrovaných zařízení. Teoreticky tedy není problém s přenositelností zařízení napříč různými zeměmi. [25, 27]

Praxe ale prozatím mírně pokulhává za oficiálními prohlášeními společnosti. Tím, že je budování celé infrastruktury založené na možnostech mobilních operátorů, jsou prozatím pokrývány pouze oblasti, kde již má tato technologie své zákazníky. Bohužel, původní předpoklady o poptávce koncových zařízení se ukázaly jako nadhodnocené, takže nevzniká takový tlak na pokrytí dalších oblastí. Kvalitní pokrytí se týká převážně městských aglomerací, plošné pokrytí horských oblastí nebo venkova je prozatím, podle některých názorů, značně diskutabilní. Navíc, díky nezájmu operátorů v mnohých státech, jsou tyto země stále zcela bez jakékoliv konektivity sítě Sigfox. V tomto ohledu má Česká republika, oproti svým zahraničním sousedům, stále nadprůměrné pokrytí svého území. [28]

3.3.2 Specifikace

Sít je celulárního charakteru s hvězdicovou topologií sítě. Data ze všech rádiových bran Sigfoxu putují nejprve do centrálního cloudu a odtud teprve ke koncovým zákazníkům. Komunikace je zabezpečená AES šifrováním, s možností volby 3 úrovní zabezpečení koncového zařízení:

- Střední – uložení bezpečnostních klíčů v paměti koncového zařízení.
- Vysoké – použití chráněného paměťového sektoru.
- Velmi vysoké – použití hardwarového šifrovacího čipu.

Rádiová komunikace využívá nelicencovaných pásem 868 MHz (EU), 915 MHz (USA) a 433 MHz (Asie). Modulace signálu je založená na UNB technologii. Komunikace má odlišné parametry pro příjem a vysílání dat.

Příjem zpráv z koncového zařízení má pro Evropu šířku pásma 100 Hz, v USA 600 Hz. Nosná frekvence se pohybuje v rozsahu 868,00 MHz až 868,60 MHz. Maximální vysílací výkon koncového zařízení je omezen na 25 mW. Pro použití na území EU navíc ještě platí omezení na max. 1% využití pásma v čase. Pro modulaci je použita DBPSK modulace (diferenciální bitové klíčování fázovým posunem).

Vysílání dat z brány využívá větší šířku pásma, a to 1,5 kHz na nosných frekvencích v rozsahu 869,40–869,65 MHz. Maximální povolený vysílací výkon brány je 500 mW, při max. 10% obsazení pásma v čase.

Rychlost komunikace je v obou směrech max. 100 bitů/s. Jeden datový paket se skládá z 24 Bajtů, z čehož 12 Bajtů tvoří samotná zpráva. Celkem je možné odeslat až 140 zpráv za den. Dosah jedné brány je v zástavbě 3–10 km, na venkově díky nižší hladině rádiového šumu 30–50 km. Teoretická hranice dosahu při přímé viditelnosti mezi zařízeními je 1 000 km. [29]

3.3.3 Určování polohy – Sigfox Atlas

V lednu 2017 byla spuštěna geolokační služba Sigfox Atlas. Služba umožňuje určení polohy koncových zařízení při současném zachování nízké spotřeby. Klasické GPS přijímače mají spotřebu okolo 17–150 mW, proto se příliš nehodí pro nízkoenergetické aplikace. Navíc je signál GPS špatně zachytitelný uvnitř budov. Řešení od společnosti Sigfox proto využívá kombinaci měření síly vysílaného signálu z koncového zařízení a detekce blízkých Wi-Fi sítí. Základní verze služby (Atlas Native) nevyžaduje žádný dodatečný hardware. Po standardní výměně paketů se sítí Sigfox se vyhrazené geolokační servery snaží vypočítat polohu koncového zařízení. Základním předpokladem je, že se podaří komunikaci zachytit na několika přijímačích zároveň. Samotný sběr informací o sledovaném paketu trvá 25 sekund, následně probíhá vyhodnocení polohy a odeslání vypočtené polohy do cloudu Sigfoxu. Podle oficiální statistiky byla u 80 % požadavků určena poloha s přesností v rozmezí 1 až 10 km. Výhodou tohoto způsobu je, že nijak nezvedá spotřebu koncového zařízení, je velmi těžce zarušitelný a provozuschopný i ve vnitřních a podzemních prostorech. Jediným omezením je zde potřeba dostatečně hustého pokrytí oblasti sítí Sigfox.

Pokročilejší metodou určování polohy je metoda používající pasivní sken okolních Wi-Fi sítí (Atlas WiFi). K tomu je ale potřeba přídavný hardware. Koncové zařízení provede měření síly signálu okolních Wi-Fi sítí a na základě nastavených filtrů vybere 2 MAC adresy okolních Wi-Fi routerů. Informace o MAC adresách je přidána do odesílaného paketu. Servery Sigfoxu pošlou požadavek na určení polohy geolokačním službám Google a HERE Technologies. Zároveň je provedena analýza polohy s využitím informace o síle signálu zachycených paketů pomocí přijímačů sítě Sigfox. Následně je provedeno porovnání výsledků a nejpravděpodobnější poloha je uložena do cloudu Sigfoxu. Koncové zařízení ale není informováno o svojí poloze, pokud to nějakým způsobem nezajistí navazující zákaznická aplikace. Konkrétní údaje o přesnosti této metody jsem nenalezl, ale můžeme předpokládat, že v případě úspěšného zachycení Wi-Fi signálu, může být přesnost polohy v rozmezí 25–50 m. [14, 30, 31]

3.3.4 Poplatky

Služba je zpoplatněná podle počtů zařízení.

Tabulka 3.3: Ukázka ceníku Sigfoxu pro ČR [32]

Tarif	BASIC	PLUS	ULTRA
Cena/zařízení/rok	140 Kč	215 Kč	247 Kč
Počet uplinků za den	2	70	140
Počet downlinků za den	1 týdně	2	4
Další služby	–	Geolokace Atlas	Prémiová podpora

3.3.5 Závěr

Iniciativa Sigfoxu byla jednou z prvních úspěšných snah o popularizaci UNB technologie a zasloužila se o rozjezd mnoha komerčních projektů v oblasti IoT. Výroba hardwaru není vázaná na jednoho výrobce, jak je tomu například u LoRa zařízení, což nám dává jistou záruku dostupnosti náhradních dílů do budoucna. Jako úskalí této technologie bych viděl centralizaci celého přístupu do jediného cloudu provozovaného touto společností, bez možností jiných alternativ. Nedokáží odhadnout, jaký dopad by měl krach této společnosti na stávající infrastrukturu sítě. Jisté ale je, že by výpadek přístupového cloudu měl za následek znepřístupnění veškerých dat z koncových zařízení.

3.4 NB-IoT

Nyní bych se přesunul do mírně odlišného spektra komunikačních technologií. Následující kapitoly se zabývají nízkoeenergetickými sítěmi operujícími v licencovaných rádiových pásmech. S tím souvisí i určité změny parametrů. Obecně mají tyto sítě větší přenosovou rychlost, větší velikost přenášených zpráv, o něco kratší dosah a častokrát i větší energetický nárok na udržení rádiového spojení. Množství spotřebované energie ale nemusí být nutně větší než u předchozích typů LPWAN, protože vyšší komunikační rychlost zase zkracuje vysílací a přijímací časy.

Jednou z takových sítí je i síť NB-IoT. Jedná se o minimalistickou verzi LTE komunikačního protokolu, tak aby splňoval požadavky pro LPWAN sítě. Ve své podstatě došlo ke zredukování energetické náročnosti komunikace na úkor přenosové rychlosti, která není u těchto aplikací stěžejní a odstranění zbytečných operací, jako je např. sledování kvality signálu.

3.4.0.1 Parametry přenosu

Síť využívá licencovaných pásem 700 MHz, 800 MHz a 900 MHz v závislosti na legislativě v místě nasazení. Komunikační kanál má šířku pásma 200 kHz, což odpovídá jednomu kanálu GSM nebo taky LTE sítě. Výběr kanálu závisí na tom, v jakém režimu je daná síť provozována. Existují celkem 3 varianty:

- Stand-alone varianta – využívá volný GSM kanál.
- Guard-band provoz – provoz ve volném kanálu nevyužívaném LTE sítí.
- In-band provoz – využití jednoho z kanálů LTE sítě.

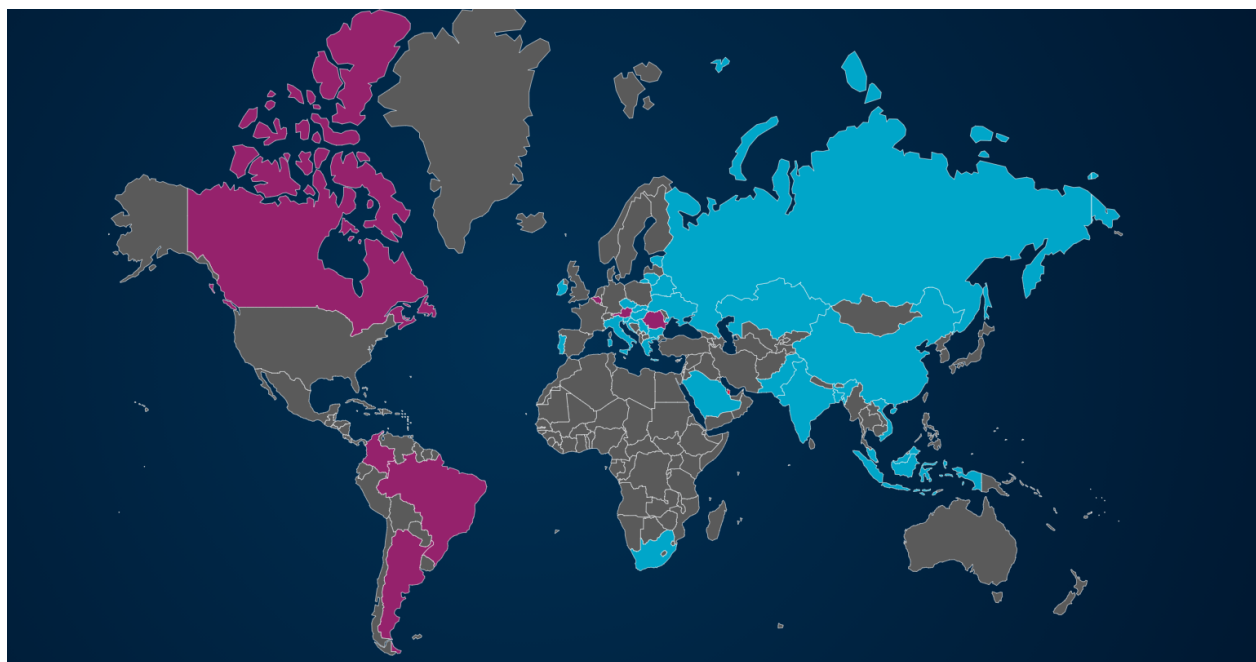
Síť má hvězdicovou topologii. Výkon vysílače koncového zařízení se pohybuje v rozmezí 25–200 mW. Dosah zařízení je až 10 km na volném prostranství, v zástavbě pak klesá na vzdálenost okolo 1 km. Data se modulují pomocí QPSK modulace. Ve srovnání s LoRa modulací je signál více náchylný na interference. Příjem dat do koncového zařízení má komunikační rychlost 20 kb/s a využívá metodu FDMA (Frequency Division Multiple Access).

Vysílání dat z koncového zařízení má rychlost 200 kb/s. Pro zajištění současného příjmu dat z více zařízení se využívá metoda OFDMA (Orthogonal frequency-division multiple access). Jedna základová stanice je schopna obsluhovat až 100 000 zařízení ve svém dosahu. Každá zpráva může obsahovat až 1600 Bajtů dat. U této sítě je potřeba počítat s poměrně vysokou latencí, okolo 1,6–10 sekund.

Podstatným omezením této technologie je nemožnost přepojování koncového zařízení mezi jednotlivými rádiovými vysílači. Po zapnutí zařízení trvá přibližně 2–5 minut, než se zařízení spojí se sítí. Od okamžiku navázání spojení je zařízení vázáno na příslušný vysílač. Nelze tedy s koncovým zařízením příliš pohybovat, protože od okamžiku navázání spojení se zařízení nemůže spojit s žádným jiným vysílačem, byť je v jeho dosahu. [33, 34]

3.4.1 Pokrytí

Zakládání a provoz potřebné infrastruktury je zcela v rukou mobilních operátorů. Na území ČR aktuálně provozují síť NB-IoT společnosti Vodafone a T-Mobile.



Obrázek 3.12: Pokrytí sítě NB-IoT [35]

3.5 LTE-M

Význam této zkratky je Long-Term Evolution Machine type communication. Můžeme se taky setkat s označením LTE Cat-M1. Vývoj této technologie začal někdy kolem roku 2014. Ke vstupu na trh pak došlo v roce 2017. Jedná se se o nízkoenergetickou odnož klasických telekomunikačních

sítí 3. a 4. generace, známějšími pod označením 3G a 4G sítí. LTE-M zároveň využívá stávající infrastrukturu 3G a 4G vysílačů, takže není potřeba budovat novou síť přístupových bran. Po technické stránce se tato síť velmi podobá síti NB-IoT. Zásadním rozdílem je možnost plynulého přepojování koncového zařízení mezi vysílači. Zařízení je tak možné použít například pro trasování zásilek. Díky podobnosti těchto dvou technologií je dokonce možné vyrábět i duální moduly, které mohou komunikovat prostřednictvím obou sítí. [34]

3.5.1 Parametry přenosu

Síť nemá striktně přidělené frekvenční pásmo, protože je vázaná na pásmo LTE sítě, které si operátoři upravují podle místních poměrů. Obecně platí, že se pohybuje někde v oblasti licencovaných pásem v rozmezí 700-900 MHz. Pro modulaci signálu je použita 16 QAM modulace v kombinaci s OFDMA. Existuje několik standardů pro tuto technologii, které se mírně liší. Standard LTE Cat M1 využívá šířku pásma 1,4 MHz, což poskytuje přenosovou rychlost až 375 kb/s v obou směrech. U standardu LTE Cat M2 byla zvýšena šířka pásma na 5 MHz. Přenosová rychlost tak dosahuje až 2,4 Mb/s pro download a 2,6 Mb/s pro upload. Tak vysoká přenosová rychlost nám umožňuje provádět na dálku aktualizace firmwaru zařízení. Tato operace se v anglické literatuře označuje zkratkou FOTA – Firmware updates Over The Air. [36]

Dalším benefitem relativně vysoké přenosové rychlosti je možnost navázat hlasový přenos s koncovým zařízením. Tato technologie se označuje zkratkou VoLTE – Voice Over LTE. V oblasti IoT se jedná o celkem nezvyklou funkcionalitu, ale i ta může najít své uplatnění. U standardu LTE Cat M1 má spojení latenci v rozmezí 100–150 ms (některé zdroje uvádějí 10–15 ms, ale tato hodnota mi nepřipadá dosažitelná), což je nesrovnatelně méně než u NB-IoT. Dokonce i navazování spojení po zapnutí zařízení je o něco rychlejší, trvá přibližně 1-2 minuty oproti 2-5 minutám u NB-IoT. [34]

U této technologie je potřeba dbát na kvalitní provedení napájení zařízení. V případě, že bude zařízení vysílat maximálním možným vysílacím výkonem 23 dBm, může špičková spotřeba zařízení přesáhnout i 500 mA

3.5.2 Pokrytí

Provoz potřebné infrastruktury zajišťují mobilní operátoři. V současné době ještě není tato technologie příliš rozšířená. I když ke vstupu této technologie na trh došlo již v roce 2017, nedošlo k takové světové expanzi, jak se původně očekávalo. Hlavní příčinou tohoto stavu byly potíže s hardwarem v letech 2018 a 2019, což vedlo k odlivu zákazníků ve prospěch sítí LoRaWAN a Sigfox.

V současné době nemá Česká republika žádného poskytovatele této sítě.

Kapitola 4

Možnosti implementace komunikačního rozhraní

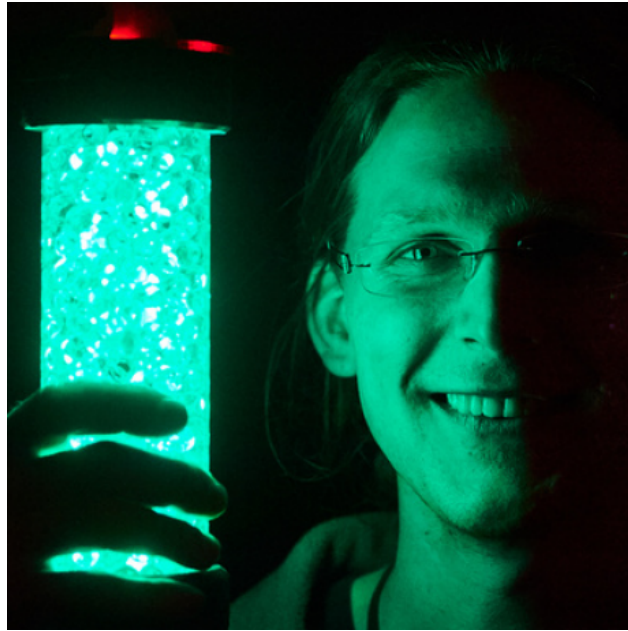
Na problematiku nasazení komunikačního rozhraní LoRa můžeme nahlížet ze dvou úhlů pohledu. Prvním aspektem, který je potřeba zohlednit, je volba fyzického zařízení pro vysílání a příjem rádiového signálu. Volba hardwaru nám následně ovlivňuje výběr softwaru – knihovny implementující pravidla pro komunikaci v pásmu sítě LoRaWAN. Některé knihovny dosahují v binární podobě velikosti i okolo 32 kB, takže nejsou vhodné pro některé typy 8 bitových mikroprocesorů s malým množstvím paměti flash.

V současné době se na trhu nejčastěji setkáváme se zařízeními v podobě vysoce integrovaných modemů, které ve většině případů neobsahují mikrokontrolér pro samotný aplikační program. Modem nejčastěji komunikuje s externím mikrokontrolérem pomocí SPI rozhraní. Konstrukce modemů a jejich signálových výstupů je v některých případech přizpůsobená rozměrům konkrétních vývojových kitů s mikrokontroléry, čímž se zjednodušuje prototypování produktu.

Vzhledem k tomu, že je technologie LoRaWAN relativní novinkou, dochází pravidelně k průběžnému vylepšování a změnám ve specifikaci komunikačního protokolu. Vzhledem ke komplexnosti celé komunikace se nejspíše nikdo nebude bez vážných důvodů zabývat vývojem vlastní knihovny na zelené louce, ale sáhne po některé z volně dostupných knihoven, implementující síťový stack sítě LoRaWAN. Mezi dva hlavní hráče na poli opensource softwaru patří *IMB LMIC framework* a *LoRa MAC Node*, podporovaný společností Semtech Corp. Pro mikroprocesory od společnosti Espressif Systems z produktové řady ESP (ESP8266, ESP32 ...) pak existují speciální knihovny, např. *ttn-esp32* nebo *ESP32_LoRaWAN*. Touto oblastí se ale nadále nebudu více zabývat, protože tato platforma není předmětem této práce. [37]

4.1 IBM LMIC framework

Velmi populární knihovnou v oblasti hobby projektů, ale i komerčních řešení se stala právě knihovna s názvem *LMiC* (zkratka *LoRa MAC in C*). Hlavním autorem projektu je Matthijs Kooijman. Práci na vývoji komunikační knihovny v jazyce C, která by implementovala komunikační protokol LoRa, zahájil někdy v polovině roku 2015. Postupem času se k vývoji softwaru připojilo dalších 6 programátorů, ale drtivou část práce odvedl právě její autor.



Obrázek 4.1: Matthijs Kooijman

Knihovna se stala velmi populární díky její snadné integraci na platformu Arduino, což přispělo k jejímu masivnímu rozšíření. V průběhu let došlo k několika změnám názvu knihovny, což vyvolalo menší vlnu komplikací pro uživatele této knihovny. Změny názvu například zapříčinili nemožnost načtení poslední verze knihovny v prostředí Arduino IDE, protože toto vývojové prostředí nepočítalo s možností přejmenování knihovny mezi jednotlivými verzemi. V roce 2017 bylo autory změněno pojmenování knihovny na *LMIC-Arduino*, následně byl projekt znovu přejmenován na *IBM LMIC framework*. Pro ještě větší zmatek je nutno podotknout, že ani v rámci diskuzních fór nedošlo k ustálení terminologie, takže je možné se v rámci diskuzí setkat ještě s variantami názvu jako *Classic LMIC* nebo taky *MCCI LMIC*. Ve všech případech se ale jedná o knihovny s původním názvem projektu *LMiC*. Zdrojové kódy projektu jsou dostupné na adrese: <https://github.com/matthijskooijman/arduino-lmic> [38, 37]

V říjnu 2020 byla oficiálně ukončena podpora a vývoj této knihovny s doporučením, aby případní zájemci přešli na knihovnu s názvem *Arduino-LMIC* (nezaměnit s knihovnou *LMIC-Arduino*!), která převzala části kódu a pokračuje zde aktivní vývoj a údržba. Je možné se setkat taky s názvem *MCCI*

LoRaWAN LMIC library. V době psaní této práce je knihovna volně dostupná na této webové stránce¹.

4.1.1 Vlastnosti knihovny

Knihovna má plnou podporu pro obsluhu vysílání v režimech Class A a Class B. Vysílání může probíhat v pásmu 868 MHz – standard pro území EU nebo v pásmu 915 MHz - standard na území USA. Autoři neprovedli plné testování softwaru, takže neposkytují žádnou záruku na bezchybnou činnost softwaru. Podle dokumentace byly otestovány následující funkcionality:

- Odesílání paketů v nastaveném intervalu.
- Šifrování a kontrola integrity zpráv.
- Příjem paketů v časovém okně RX2.
- Nastavení vysílací frekvence a komunikační rychlosti.
- Over-the-air activation (OTAA).

Autoři upozorňují, že u následujících funkcionalit neproběhlo žádné testování:

- Příjem paketů v časovém okně RX1.
- Příjem a zpracování MAC příkazů.
- Vysílání v režimu Class B.

Knihovna je určena pro široké spektru vysílacích modemů s komunikací pomocí SPI rozhraní. Součástí kódu je navíc plná implementace síťového stacku LoRaWAN, takže použití není vázané na hardware s jeho implementací na straně vysílače, jako je např. Microchip RN2483. Knihovna zajišťuje plné nízkoúrovňové řízení vysílání modemu přes SPI rozhraní. Knihovna podporuje modemy z rodiny zařízení SX1272 a SX1276, konkrétně typy: SX1273, SX1277, SX1278 a SX1279.

Knihovna nemá žádné závislosti na použité architektuře, tzn. není vázaná na Arduino framework nebo případně nějakou HAL knihovnu. Jednoduše tedy můžeme přecházet mezi různými platformami a architekturami, konkrétně mikroprocesory AVR nebo ARM, případně platformou ESP. Autoři potvrzují funkčnost v kombinaci se zařízeními: Arduino Uno, Pinoccio Scout, Teensy LC, 3.x, ESP8266 a Arduino 101.

¹<https://github.com/mcci-catena/arduino-lmic>



Obrázek 4.2: Modem osazený čipem Semtech SX1276

Při volbě hardwaru je důležité vzít v potaz velikost softwaru knihovny, která v plné velikosti dosahuje okolo 32 kB paměti flash. Pokud uspěje snaha autorů o nahrazení algoritmu AES šifrování, bude možné zmenšit velikost výsledného binárního souboru o cca 8 kB.

4.2 Arduino-LMIC

Od října 2020 lze považovat tuto knihovnu jako referenční implementaci LoRa protokolu na 8bitových a 32bitových mikroprocesorech. Knihovna je vhodná spíše pro pokročilé programátory, protože obsahuje větší množství nízkoúrovňových metod, tudíž ke korektní aplikaci je potřeba pečlivě nastudovat přiložený manuál. Pro méně zdatné programátory byla vytvořena uživatelsky přívětivější varianta knihovny, s množstvím vysokoúrovňových metod. Lze ji dohledat pod názvem *MCCI Arduino LoRaWAN* nebo taky na této webové stránce².

4.2.1 Vlastnosti

Knihovna podporuje vysílání a příjem v režimech Class A a Class B. Proti IBM LMIC frameworku má daleko širší regionální pokrytí vysílacích pásem. V současné době má plnou podporu pro vysílání na těchto kanálech:

- EU-868 – pásmo 868 MHz.
- US-915 – pásmo 915 MHz.
- AU-921 – pásmo 921 MHz.
- AS-923 – pásmo 923 MHz.
- IN-866 – pásmo 866 MHz.

²<https://github.com/mcci-catena/arduino-lorawan>

Regionální předpony jako EU, US atd. jsou spíše obecným označením daného pásma než geografickou zkratkou. V závislosti na frekvenčním plánu dané země je častokrát možné vysílat hned na několika pásmech. Aktuální informace je možné získat na stránkách Lora Alliance nebo v dokumentu *RP002-1.0.2 LoRaWAN® Regional Parameters*. [39]

Vzhledem k tomu, že stále probíhá aktivní vývoj této knihovny, nemá cenu příliš zabíhat do detailů, které budou nejspíše za krátko neaktuální. Použití současné nejnovější verze vyžaduje pečlivé odladění časování přijímání a vysílání intervalů. Samotná knihovna tohle prozatím nijak negarantuje a je na programátorovi, aby zajistil dostatečně precizní načasování jednotlivých operací. Více informací najdete v poznámkách k projektu. [40]

4.3 LoRa MAC node

Oproti předchozím knihovnám je vývoj této softwarové knihovny více konzistentní. Nedocházelo v průběhu vývoje k živelným změnám verzování, přejmenovávání atd. Lze tedy očekávat stabilní vývoj a implementaci nových standardů sítě LoRaWAN i v horizontu několika příštích let. Projekt se v začátcích inspiroval projektem mbed.org. Zdrojové kódy projektu jsou volně dostupné na webovém portálu Github, co je ale hlavní, komunita udržuje kvalitní dokumentaci celého projektu na internetové adrese <http://stackforce.github.io/LoRaMac-doc/>. Jelikož se projekt nesoustředí pouze na samotnou implementaci síťového stacku, ale jeho součástí jsou i rozličné drivery pro různý podpůrný hardware, je tato dokumentace klíčovým bodem k pochopení celé struktury knihovny.

Autoři projektu se snaží udržet krok s novými standardy sítě LoRaWAN, které postupně zapracovávají do nových verzí softwaru. V době psaní této práce byla poslední stabilní verze knihovny shodná se specifikací LoRaWAN L2 1.0.3, přičemž momentálně probíhá vývoj verze, která bude odpovídat specifikaci LoRaWAN L2 1.0.4 / 1.1.1.



Obrázek 4.3: Partneři projektu

4.3.1 Vlastnosti

Knihovna nabízí daleko širší možnosti využití než předchozí zmíněné varianty. Jako jedna z mála knihoven podporuje všechny 3 vysílací režimy – Class A, B i C. Knihovna zřejmě prochází i průběžným testováním (domněnka), protože v celé dokumentaci projektu jsem nenarazil na žádné výstrahy o možné nefunkčnosti, s výjimkou sekce hlášení nalezených chyb. Samozřejmě i zde se autoři kryjí zřeknutím se zodpovědnosti za škody způsobené použitím tohoto softwaru. Knihovna je distribuována pod revidovanou BSD licencí.

Knihovna obsahuje podporu pro vysílání na daleko širším počtu kanálů než předchozí knihovny.

- EU868 – Evropa 863–870 MHz.
- US915 – Severní Amerika 902–928 MHz.
- CN779 – Čínská lidová republika 779–787 MHz.
- EU433 – Evropa 433 MHz.
- AU915 – Austrálie 915–928 MHz.
- CN470 – Čínská lidová republika 470–510 MHz.
- AS923 – pásmo 923 MHz (v drobných odchylkách frekvence platí pro Indonésii, Vietnam).
- KR923 – Jižní Korea 920–923 MHz.
- IN865 – Indie 865–867 MHz.
- RU864 – Ruská federace 864–870 MHz.

Z hlediska vysílacího hardwaru, je možné použít knihovnu pro všechny modemy s čipy: SX1272, SX1276, SX126x, LR1110. Po softwarové stránce má knihovna solidně zavedenou hierarchii, takže se v ní dobře orientuje. Pro mne důležitou částí dokumentace je návod k vytvoření portu knihovny na jinou hardwarovou platformu. Autoři zde uvádí seznam konkrétních zdrojových kódů, které je potřeba modifikovat a doimplementovat, aby bylo možné přidat do projektu podporu pro nový mikrokontrolér.

Kapitola 5

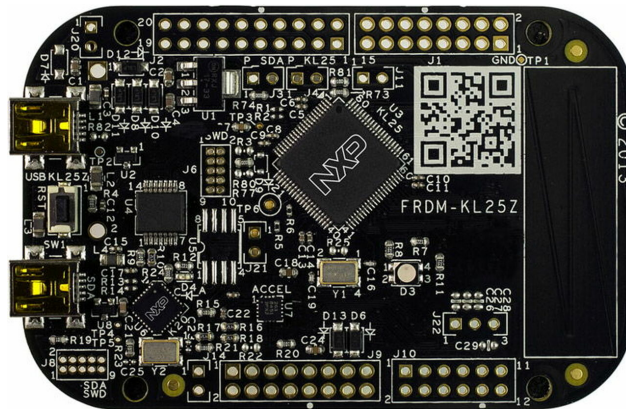
Portace softwarové knihovny sítě LoRa-WAN na vybraný mikrokontrolér

Jako základní kámen pro budoucí knihovnu byla zvolena knihovna s názvem LoRa MAC node. Konkrétně její verze s číslem 4.4.4. z vývojové větve s názvem „master”. Knihovna implementuje specifikaci LoRaWAN® protokolu s pořadovým číslem 1.0.3. V době psaní této práce byla již vydaná nová směrnice s pořadovým číslem 1.0.4 a 1.1.0, ale potřebná knihovna síťového stacku je teprve ve fázi vývoje, proto jsem zvolil nejvyšší stabilní verzi.

Cílovou hardwarovou platformou je vývojová deska FRDM-KL25Z od firmy Freescale. Mimo jiné se používá pro výukové účely na VŠB-TUO. V této souvislosti zároveň existuje programová šablona od Ing. Martina Stankuše, Ph.D., která usnadňuje práci s registry a periferiemi na tomto mikrokontroléru. Vzhledem k tomu, že bude výsledná knihovna sloužit nejspíše jen k použití s výše uvedenou vývojovou deskou, bylo rozhodnuto použít již hotových maker a konstant z šablony. Zvýší se tím srozumitelnost kódu pro ostatní programátory, kteří již nejspíše budou mít určité zkušenosti s touto šablonou z výuky na VŠB-TUO.

Pro snadné propojení modemu sítě LoRaWAN s mikrokontrolérem, byl jako rádiový modem zvolen SX1261MB2BAS. Jak už vyplývá z názvu, jedná se o typ osazený integrovaným obvodem SX1261. Důvodem pro tuto volbu byla jeho kompatibilita se signálovými vstupy a výstupy z FRDM-KL25Z. Oba výše uvedené moduly je možné jednoduše propojit pouhým zasunutím jednoho plošného spoje do patice druhého modulu.

5.1 Vývojový kit FRDM-KL25Z



Obrázek 5.1: Vývojová deska FRDM-KL25Z

Tento vývojový kit obsahuje kromě 32bitového mikrokontroléru i OpenSDA debugger, který usnadňuje ladění programu mikrokontroléru. Na vývojové desce se nachází mikroprocesor KL25Z128VLK s jádrem ARM® Cortex™-M0+. Maximální takt tohoto mikroprocesoru je 48 MHz. Programová paměť flash má kapacitu 128 kB. Dále bych zde chtěl uvést některé zajímavé periferie a funkcionality tohoto mikrokontroléru.

- DMA¹.
 - Umožňuje předávání dat mezi periferiemi bez potřeby přímé účasti jádra, což usnadňuje obsluhu asynchronních událostí, třeba dokončení převodu AD převodníku.
- Nízkoenergetické kapacitní snímání dotyku.
 - Lze využít např. k vytvoření kapacitního zapínací tlačítka, které probudí mikroprocesor z režimu spánku.
 - Vývojový kit využívá tuto funkcionalitu k vytvoření kapacitního dotykového posuvníku.
- 12 nebo 16 bitový A/D převodník s možností diferenciálního měření.
- 12 bitový D/A převodník.
- 3 sběrnice UART s podporou DMA a asynchronního vysílání v režimu spánku.
- 2 sběrnice I²C.
- I²S sběrnice.

¹Direct memory access

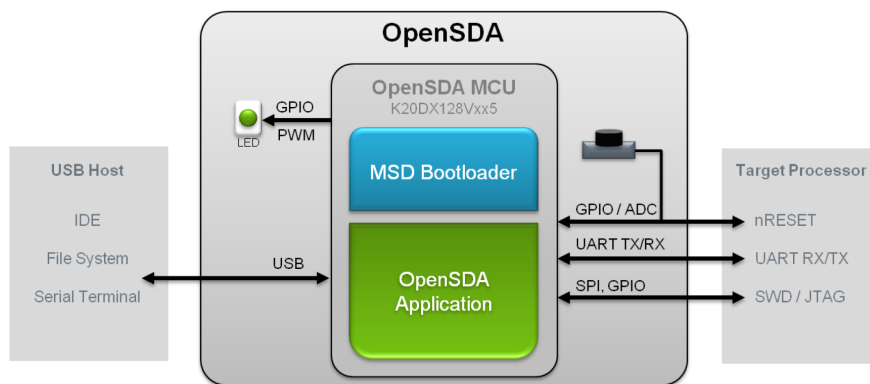
- Full-speed USB OTG.
 - Interní watchdog.
 - RTC obvod.
- U použitého vývojové kitu bohužel není hodinový vstup oscilátoru osazen krystalem.

Z hlediska nízkoeenergetických aplikací je tento mikroprocesor zajímavý nejen díky nízké spotřebě, ale i možnostem, které umožňují minimalizovat spotřebu elektrické energie u nevyužívaných periférií. Dle údajů od výrobce má tento mikroprocesor o 50 % menší spotřebu než 8 nebo 16 bitové mikroprocesory se srovnatelným výpočetním výkonem. Další úspory energie je možné dosáhnout použitím spánkových režimů. Díky technologii DMA je možné vykonávat některé operace bez přímé účasti jádra, které může být po celou dobu v režimu spánku s minimální spotřebou. Lze také selektivně vypínat periferie a některé vnitřní sběrnice, nebo rovnou zastavit zdroj hlavního hodinového signálu, zatímco vybrané periferie poběží dál z alternativního zdroje hodinového signálu. [41]

Vývojový kit je možné napájet hned 5 různými způsoby, ale pouze při napájení z OpenSDA USB portu je možné použít debugger, v ostatních případech není tento obvod napájen.

Tabulka 5.1: Možnosti napájení vývojové desky

Zdroj napájení	Povolené napětí	Aktivní OpenSDA
OpenSDA USB	5 V	Ano
KL25Z USB	5 V	Ne
V _{IN} pin	4,3–9 V	Ne
3,3 V pin	1,7–3,6 V	Ne
Knoflíková baterie	1,7–3,6 V	Ne

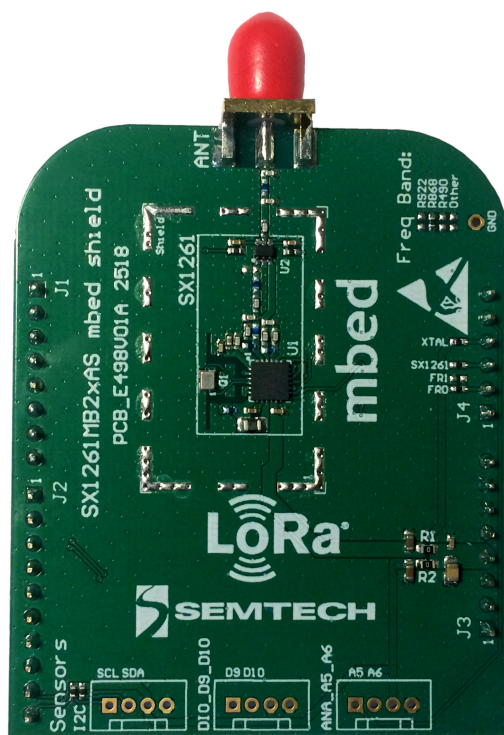


Obrázek 5.2: Blokový diagram debuggeru

Debugovací rozhraní se skládá z mikroprocesoru řady Freescale Kinetis K20, který zprostředkovává USB rozhraní se standardem OpenSDA. Druhá strana komunikace s hlavním mikroprocesorem probíhá pomocí SWD rozhraní. Navíc je možné vytvořit spojení mezi PC a hlavním mikrokontrolérem pomocí UARTu, který je připojený k debuggeru. V počítači se potom tato sériová linka zobrazí jako zařízení připojené na COM portu počítače. Vývojový kit je navíc ještě osazen RGB LED diodou a akcelerometrem připojeným k I²C sběrnici. LED diodu zužitkujeme při signalizaci stavů programu. [42]

5.2 Modem s SX1261

V této aplikaci byl použit modem založený na integrovaném obvodu SX1261. Jedná se o univerzální sub-gigahertzový vysílač pro přenosy na velké vzdálenosti. Integrovaný obvod se vyrábí ve dvou velmi podobných variantách, lišící se pouze vysílacím výkonem. Model SX1261 dosahuje maximálního vysílacího výkonu 31.6 mW (+15 dBm) a model SX1262 až 158.5 mW (+22 dBm). Modem je navržen pro použití v zařízeních s bateriovým napájením, tudíž zde byl kladen důraz na velmi nízkou spotřebu. Zařízení je koncipováno jako univerzální vysílač/přijímač pro různé nízkoenergetické aplikace v pásmu frekvencí od 150 MHz do 960 MHz. Volba vysílacího pásma a vysílacích výkonů musí samozřejmě respektovat legislativu země, ve které bude zařízení provozováno.



Obrázek 5.3: Modem SX1261

Tabulka 5.2: Doporučené provozní podmínky modemu SX1261

Doporučené provozní podmínky			
Veličina	Min	Max	Jednotka
Napájecí napětí	1,8	3,7	V
Teplota	-40	85	°C
Kapacitní zátěž dig. výstupů	-	20	pF
Vstupní rádiový signál	-	10	dBm

Zařízení má plnou podporu pro LoRa modulaci a splňuje všechny požadavky na fyzickou vrstvu sítě LoRaWAN, podle specifikace vydané organizací LoRa Alliance. Dále je možné zařízení provozovat v režimu GFSK modulace (Gaussian frequency-shift keying – Gaussovské klíčování frekvenčním posunem). Pracovní frekvence modemu je dána krystalovým oscilátorem o frekvenci 32 MHz. Z této frekvence je poté syntetizován nosný kmitočet vysílače. Výrobce dává záruku, že odchylka základní frekvence oscilátoru od nominální hodnoty nepřesáhne ± 30 ppm po celou dobu životnosti zařízení při dodržení provozních podmínek.

Tabulka 5.3: Mezní provozní hodnoty modemu SX1261

Mezní hodnoty zařízení			
Veličina	Min	Max	Jednotka
Napájecí napětí	-0,5	3,9	V
Teplota	-55	125	°C
Vstupní rádiový signál	-	10	dBm

5.3 Vnitřní struktura knihovny LoRa MAC node

Díky snaze autorů o pokrytí širokého spektra hardwaru má tato knihovna relativně členitou strukturu. Aby bylo možné jednoduše rozšiřovat knihovnu o další hardware, vytvořili autoři hned několik úrovní abstrakce softwaru, které do sebe postupně zapadají a skládají se do jednoho celku. Kromě toho se autoři rozhodli přibalit do knihovny i menší knihovny pro vybrané hardwarové komponenty. Nejspíš se jedná o různé periferie vývojových kitů, které jsou v seznamu podporovaných zařízení. Tato skutečnost mi trochu komplikovala práci a ztěžovala mi identifikaci klíčových prvků. Hlavně v začátcích nebylo vůbec jednoduché určit, co mohou zamýšlené změny v knihovně způsobit v rámci celku.

Z těchto důvodů jsem se rozhodl neměnit zbytečně definice funkcí. Tím se minimalizuje počet potřebných úprav napříč programem. V případě, že autoři původní knihovny vydají novou verzi

knihovny, která bude implementovat nové verze standardu LoRaWAN, mělo by být možné provést sloučení této nové verze s mnou upravenou verzí na mikrokontrolér KL25Z. V pár případech jsem byl sice takto nucen zachovat z praktického hlediska zbytečně komplikovaná volání funkcí, která v současném kontextu nedávají moc smysl, ale zachová se tím zpětná kompatibilita.

5.4 Provedené úpravy

V následujících kapitolách jsou popsány změny, které bylo potřeba provést k tomu, aby bylo možné knihovnu použít pro výše zmiňovaný 32bitový mikroprocesor. Většina úprav spočívala v definování obslužných funkcí, které zprostředkují obsluhu hardwarových periférií. Pro práci s perifériemi jsem nepoužil žádnou HAL knihovnu, ale využíval jsem přímého zápisu do registrů. Pro adresaci registrů jsem použil knihovny od doktora Stankuše, která obsahuje definice adres jednotlivých registrů. Program je tím pádem čitelnější a srozumitelnější.

Pro každou funkcionalitu, kterou bylo potřeba přeprogramovat, jsem napsal krátký testovací program. Cílem těchto testů je ověřit, zda některá z provedených úprav neobsahuje programovou chybu. Návrh testovacích scénářů jsem prováděl metodou testování bílé skříňky. V některých případech ale nebylo možné otestovat všechny možné kombinace nastavení, protože byl počet možných kombinací příliš velký na to, aby se dal test vyhodnotit v rozumném čase. Vzhledem k tomu, že se jedná o programy vykonávané mikrokontrolérem, je k vyhodnocování výsledku testu potřeba osciloskop a debugger pro 32bitové mikrokontroléry. Jedině tak je možné ověřit, že zařízení funguje podle očekávání. Jedná se tedy o manuální testování.

Při návrhu testovacích programů jsem se snažil minimalizovat množství importů z jiných knihoven, aby byl test minimálně závislý na funkcionalitě jiné komponenty knihovny. Testovací programy se nachází ve složce `LoRa_node/tests`. Pokud budou testy prováděny postupně podle číselného pořadí v názvu testu, nemělo by dojít k situaci, kdy výsledek testu bude indikovat chybu, jejíž příčina bude způsobena chybou v importované knihovně, která nesouvisí s aktuálním testovacím scénářem. Pokud se projeví v některém z testů chyba (chování mikrokontroléru neodpovídá parametrům testu), není dobré pokračovat v navazujících testech – jejich výsledek může být ovlivněn právě zjištěnou chybou.

5.5 Obsluha GPIO

Z knihovny jsem odstranil podporu IO expandéru typu SX1509 s I²C rozhraním. V této aplikaci jej nebude potřeba. Z tohoto důvodu již není potřeba definovat konstantu `BOARD_IOE_EXT` v parametrech kompilátoru. Pokud bude přesto definována, nebude to mít na program žádný vliv.

Z hlediska obsluhy přerušení, při změnách logické hodnoty pinu, byla zachována původní deklarace všech funkcí. Při použití tohoto mikrokontroléru je potřeba mít na zřeteli, že je možné generovat hardwarové přerušení pouze na pinech 0, 1, 4, 5, 12, 13, 16 pro port A a 0, 1, 2, 3, 4, 5, 6,

7, 13 pro port D. Navíc je možné celkem elegantně svázat vybrané hardwarové přerušení od vstupu, s konkrétní callback funkcí.

Při definování konstant pro řízení RGB LED na vývojovém kitu jsem narazil na menší zádrhel. Výrobce vývojového kitu celkem nešťastně propojil ovládání modré LED s pinem PD_1 na pinové liště kitu. Shodou náhod je na tento pin vyveden hodinový signál SPI komunikace s LoRa modemem. Stručně řečeno – při použití tohoto vývojového kitu a LoRa modemu bude vždy modrá LED blikat v rytmu SPI sběrnice. Aby nedocházelo k celkem zákeřné chybě, nenajdete v definici konstant konstantu pro ovládání modré složky RGB diody. Lze blikat pouze se zelenou a červenou LED diodou.

5.6 Sériová komunikace

Mikrokontrolér má hardwarovou podporu 2 SPI rozhraní a 2 UART rozhraní. Pro komunikaci s LoRa modemem je potřeba pouze SPI rozhraní. Vzhledem k odlišnostem obsluhy SPI rozhraní byla tato část naprogramována úplně od začátku. Největším oříškem bylo pro mne vytvoření funkce, která určí vhodnou kombinaci nastavení dvou kaskádních děliček hodinového signálu, aby sériová linka měla požadovanou komunikační rychlost. Z programátorského hlediska je pohodlnější zadat v nastavení SPI požadovanou komunikační rychlost, než při každé změně počítat na kalkulátoru potřebný dělicí poměr. Původně jsem se snažil najít rovnici, jejímž řešením bude požadovaná kombinace dělicích poměrů, ale zde jsem se nedobral výsledku. Vzhledem k tomu, že existuje pouze 72 možných kombinací, je možné správnou kombinaci najít dostatečně rychle i zkoušením všech kombinací. Pokud programátor zadá nedosažitelnou komunikační rychlost (tj. takovou, která není celočíselným násobkem frekvence taktu mikroprocesoru), funkce vyvolá makro `assert(FAIL)`, což signalizuje špatné vstupní parametry funkce.

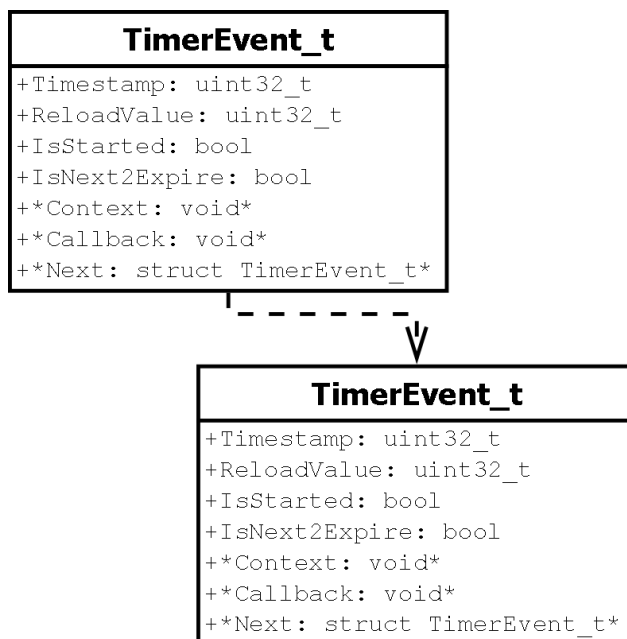
Pro svoji potřebu jsem v knihovně naprogramoval i ovladač pro UART rozhraní. Není sice důležitý pro samotné vysílání pomocí LoRaWAN modemu, ale je možné jej využít ke komunikaci s počítačem pomocí USB rozhraní. Stejného principu se využívá u platformy Arduino, v podobě známé knihovny `Serial`. Rozhraní UART0 je propojeno s debuggovací sondou, a ta dále přenáší komunikaci na USB rozhraní vývojového kitu, viz obrázek č. 5.2.

5.7 Plánovač úloh

V rámci úprav této knihovny bylo potřeba vyřešit problém se zpracováním asynchronních událostí. Existuje zde hned několik situací, ve kterých je dokončení operace podmíněno uplynutím předem známého časového intervalu. Použití čekací smyčky zde nepřipadá v úvahu, protože se jedná o jednodířkové zařízení – došlo by k zbytečnému zablokování ostatních výpočtů. Bylo potřeba použít přerušovací systém mikrokontroléru v kombinaci s programovatelným časovačem. Aby ale každá

událost nepotřebovala separátní hardwarový časovač (což by ani nebylo možné), je potřeba řadit jednotlivé události do fronty.

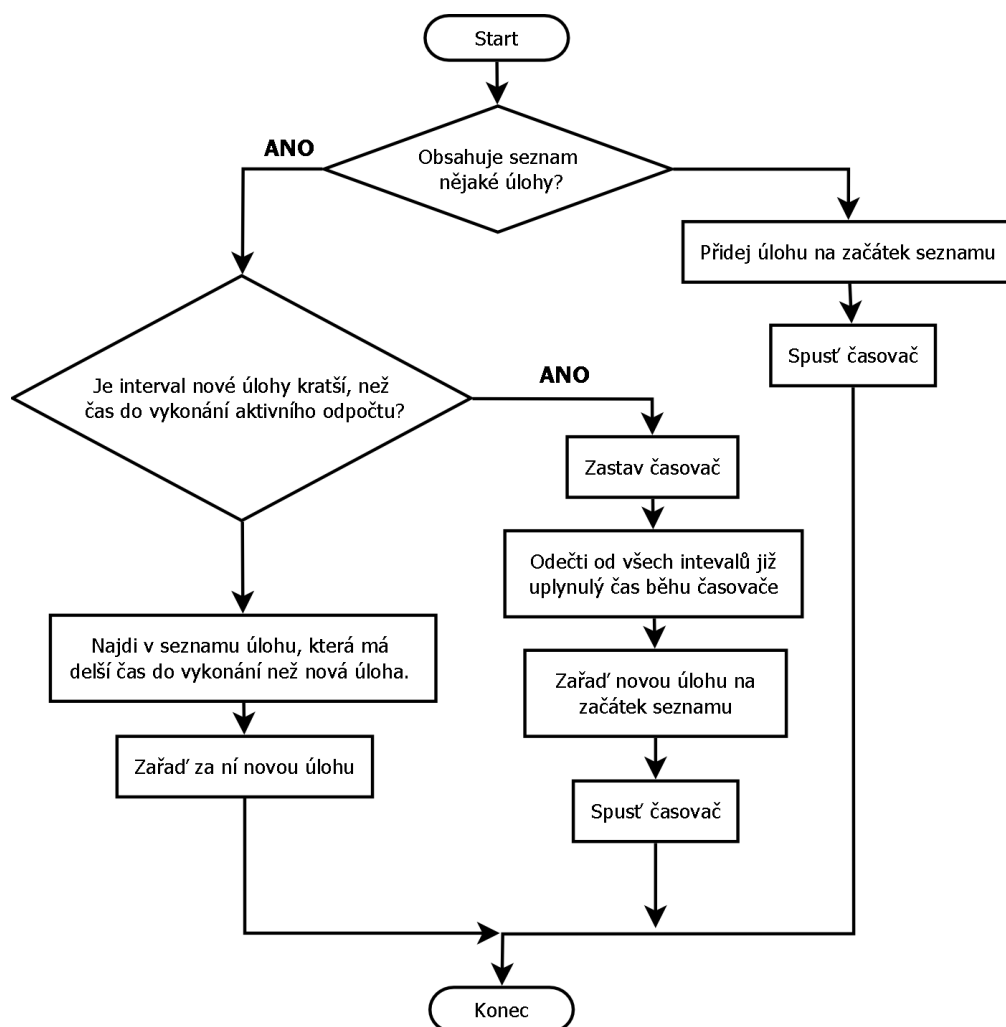
Původní řešení (knihovna LoRa MAC node) tohoto problému využívalo obvodu reálného času v kombinaci s generováním hardwarového přerušení. V této aplikaci je ale potřeba se obejít bez obvodu reálného času. Proto jsem jako časovací mechanismus zvolil periodický časovač PIT (periodic interrupt timer). Pro zachycení informací o všech událostech čekajících na zpracování slouží spojový seznam obsahující struktury s datovým typem `TimerEvent_t`. Definici datové typu této strukturu můžeme vidět na obrázku č. 5.4.



Obrázek 5.4: Datová struktura pro plánovač

Informace o době čekání do vykonání úlohy je uložena v proměnné `ReloadValue` a `TimeStamp`. Hodnota `TimeStamp` se postupně snižuje s každou aktualizací spojového seznamu, kdežto proměnná `ReloadValue` uchovává nezměněnou informaci o požadovaném zpoždění od zařazení úlohy ke zpracování do vlastního vykonání operace. Tuto informaci je potřeba uchovat pro případ opětovného spuštění úlohy. V proměnné `Callback` se nachází ukazatel na metodu, která se má vykonat po uplynutí nastaveného intervalu. Této metodě můžeme předávat parametry prostřednictvím ukazatele `Context`. Další úloha v pořadí se nachází v ukazateli `Next`.

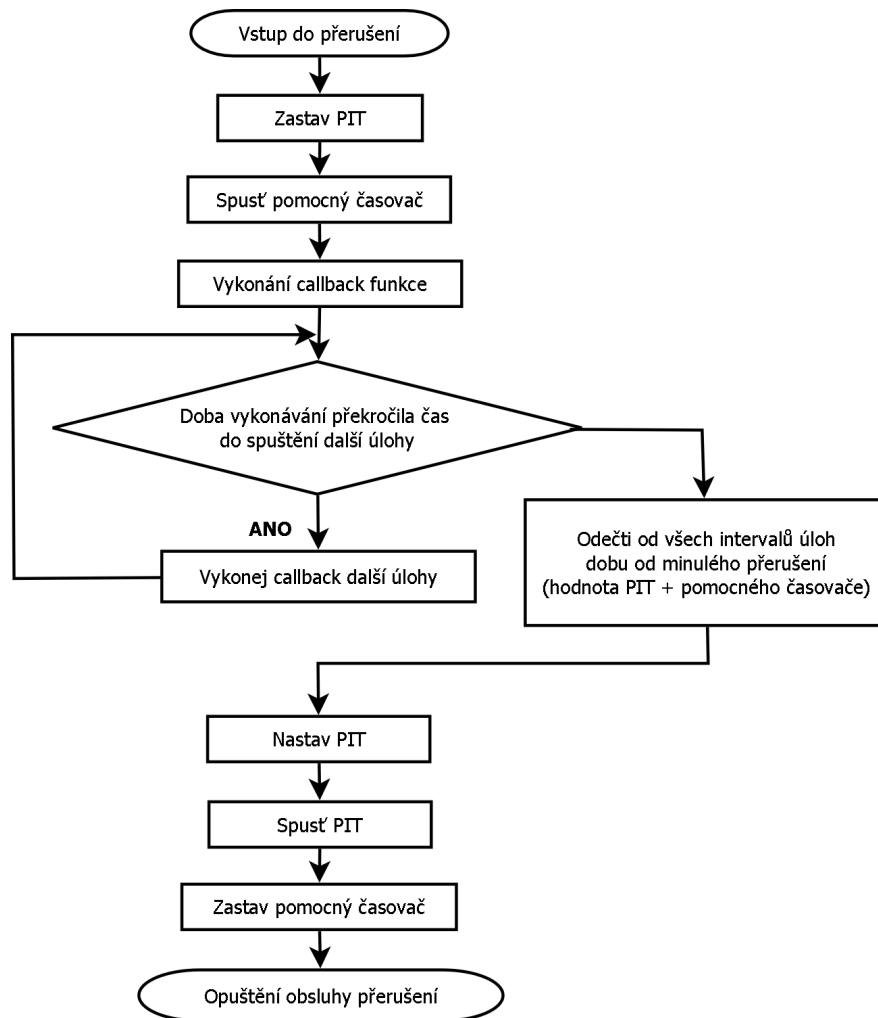
Zařazení nové úlohy do spojového seznamu má následující průběh.



Obrázek 5.5: Přidání nové úlohy do plánovače

Pokud je spojový seznam prázdný, vloží se na jeho začátek struktura s informacemi o nové úloze a spustí se PIT časovač. Doběhnutí časovače vyvolá hardwarové přerušení a spustí další operace. V případě, že se již ve spojovém seznamu nachází jiné úlohy je potřeba správně zařadit novou úlohu. Jestliže nová úloha má být vykonána dříve, než doběhne aktuálně běžící odpočet do dalšího přerušení, musí se časovač zastavit a zahájí se odpočet kratšího času od nové úlohy. Pokud má nová úloha delší čas do vykonání, zařadí se na příslušné místo spojového seznamu.

První úloha ve spojovém seznamu má vždy nejkratší interval do vykonání a tento interval se po přepočtu na počet tiků časovače zapisuje do příslušného registru PIT. Doběhnutí časovače vyvolá přerušení a spustí vykonání přiřazené funkce.



Obrázek 5.6: Obsluha přerušení PIT

V přerušení dojde k vykonání `Callback` funkce. Zároveň je ale potřeba zohlednit dobu vykonávání této metody, protože během zpracovávání už možná mělo dojít k vykonání další úlohy v pořadníku. Poté, co již jsou odbaveny všechny expirované události se provede nastavení nového intervalu do PIT a zastavení pomocného časovače. Pomocný časovač slouží k měření času od vstupu do obsluhy přerušení.

Tímto algoritmem je možné zpracovat i velké počty čekajících operací. Důsledkem velkého počtu takovýchto asynchronních operací bude ale přirozeně zpomalení vykonávání hlavní programové smyčky, protože bude program častěji obsluhovat přerušení. Zde je nutno mít na paměti, že každé zpracování přerušení konzumuje navíc nemalé množství procesorového času.

Kapitola 6

Testování softwaru a zkušební přenos dat

Následující odstavce popisují funkcionalitu testovacích programů, které jsem průběžně vytvářel při úpravě knihovny. Zdrojové kódy nejsou součástí tohoto dokumentu, proto je následující kapitola spíše jen dokumentačního rázu, aby existovala nějaká dokumentace k těmto testovacím programům a způsobům vyhodnocení.

6.1 Ovládání GPIO

Prvním testovacím scénářem bylo ověření správnosti obsluhy GPIO pinů a zachycení vnějšího hardwarového přerušení při změně logické hodnoty pinu. K tomuto účelu slouží testovací program s názvem `01_gpio_test.c`. K vyhodnocení tohoto testu je potřeba ladicí software, aby bylo možné ověřit skutečné stavy registrů mikrokontroléru. Program má následující funkcionalitu. Pin PD1 a PB19 se nastaví jako výstupní piny. Každých 100 ms se změní logická hodnota výstupního pinu, ale každý z pinů je ovládán jiným druhem funkce. Pin PA1 by měl být nastavený jako vstupní pin s aktivním pull up rezistorem a při sestupné hraně signálu generovat postupně 5 krát hardwarové přerušení. Po 5. přerušení by mělo být hardwarové přerušení deaktivováno a dále by již neměl program reagovat na vstupní signál. Pin PA4 by měl být nastavený jako vstupní pin. Jeho logická hodnota se periodicky ukládá do proměnné `pinValue`. Tento test proběhl úspěšně.

6.2 SPI komunikace

2. testem je zkouška SPI sběrnice. Knihovna provede inicializaci rozhraní SPI1 (klidový stav hodin log. 1, data na náběžné hraně hodinového signálu). Následně je nastavena komunikační rychlost na 125 kb/s. Funkce `SpiFrequency` obsahuje kontrolu nastavované komunikační rychlosti. Pokud se programátor pokusí nastavit komunikační rychlost sběrnice, která není dosažitelná použitím žádné z kombinací nastavení 2stupňové děličky hodinového signálu, dojde k zavolání funkce `assert`, která zablokuje další chod programu. V případě správné inicializace sběrnice začne program posílat vždy

2 krát po sobě 8bitové číslo, počínaje hodnotou 0xAA. Každých dalších 100 ms se navyšuje hodnota odesílaného čísla o jedničku a opakuje se vysílání. Test proběhl úspěšně.

6.3 UART komunikace

3. testem je zkouška UART rozhraní. Rozhraní UART0 je inicializováno na komunikační rychlost 115,2 kb/s, žádné parita, 1 stop bit. Následně začne mikrokontrolér odesílat různé testovací textové řetězce s využitím různých funkcí (funkce pro odesílání pole znaků, 16bitových čísel, 32bitových čísel atd.). Konkrétní hodnoty najdete ve zdrojovém kódu programu. Toto rozhraní slouží pouze pro ladicí účely, není nezbytně nutné pro vysílání pomocí LoRa modemu. Data ze sběrnice UART0 odesílá ladicí sonda na USB rozhraní kitu, kde je můžeme zachytit pomocí libovolného programu, v mém případě programu Putty. Test této sběrnice dopadl úspěšně.

6.4 Zkouška časování

4. test je kontrola funkčnosti knihovny RTC. Knihovnu jsem byl nucen přeprogramovat a jako časovou referenci zvolit časovač SysTick. Z důvodu zpětné kompatibility s knihovnou LoRa MAC Node jsem ale zachoval původní názvy funkcí, i když se pro měření času nepoužívá obvod reálného času. Program vyčítá v pravidelných intervalech čas z této knihovny funkce. Testování proběhlo tak, že jsem nechal program běžet po dobu 10 minut a pomocí ladicího programu Ozone jsem sledoval hodnoty proměnných s časem. I po 10 minutách měření se shodoval čas vyčtený z mikrokontroléru s časem zaznamenávaným v programu Ozone.

6.5 Plánovač událostí

5. test spočíval v otestování činnosti plánovače událostí. Cílem testu bylo zjistit, zda program správně řadí asynchronní události do fronty a vykonává je po uplynutí nastaveného času. Tato funkcionality se využívá při plánování vysílání paketů, kdy se v programu vytvoří záznam o provedení určité funkce s nastaveným zpožděním. Aby program nemusel zbytečně blokovat ostatní výpočty stáním v čekací smyčce, je tento odpočet času realizován hardwarovým časovačem generujícím přerušení. Testovací program vytváří neustále 3 druhy událostí s odloženým vykonáním o 10, 50 a 200 ms. Vyvolání každé callback funkce indikuje změna logické hodnoty přiřazeného pinu mikrokontroléru. Toto dynamické chování programu není možné testovat pomocí ladicí sondy. K vyhodnocení testu jsem potřeboval vícekanálový osciloskop, abych mohl ověřit souběh jednotlivých callback funkcí.

6.6 Test LED indikátorů

6. test provádí otestování pomocné knihovny `progress_bar.h`, která slouží pro ovládání signálních LED diod na mnou vytvořeném přípravku. Jedná se pouze o pomocnou knihovnu, která mi usnadnila signalizaci stavů mikrokontroléru.

6.7 Komunikace s modemem

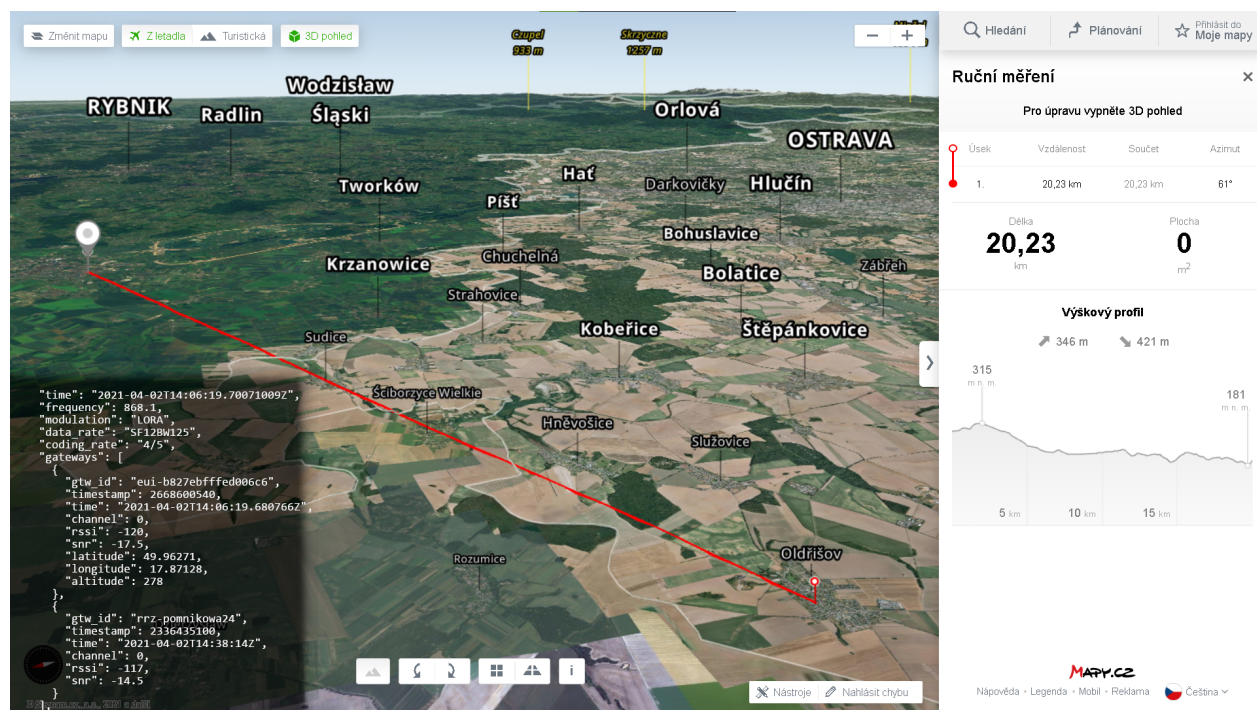
7. test provádí test komunikace mikrokontroléru s LoRa modemem SX1261. Program provede inicializaci SPI rozhraní a následně pomocí této sběrnice inicializuje i modem, který provede svou diagnostiku. Následuje vyčtení chybového registru modemu a vypsání stavu všech 8 hardwarových komponent. Vyčítání chybového registru se opakuje každou vteřinu. K úspěšnému provedení tohoto testu vedla celkem dlouhá cesta, která trvala několik týdnů. Protože se jednalo o jeden z prvních testů, ve kterém byla importována značná část celého softwaru, došlo k přetečení stacku mikrokontroléru, několika chybám softwaru, které byly způsobeny překlepy v knihovnách a podobným problémům. Po opravení všech odhalených chyb se ale stále nedařilo inicializovat modem. Testovací program neustále indikoval selhání všech hardwarových komponent modemu. Po hloubkové kontrole knihovny se ukázalo, že měla knihovna špatně nastavené parametry komunikace po SPI sběrnici. Klidový stav hodinového signálu a polarita hodin se neshodovaly s nastavením uvedeným v datasheetu modemu. Opravení nastavení komunikace odstranilo všechny indikované chybové stavy a modem začal komunikovat s mikrokontrolérem. Test tedy proběhl úspěšně.

6.8 Zkušební spojení s LoRaWAN bránou

K provedení zkušebního rádiového spoje bylo potřeba vytvořit poměrně komplexní program. Byť je značná část síťových operací již zautomatizovaná v samotné knihovně, stále zbývá mnoho záležitostí, které musí programátor ošetřit vlastní aplikací. Při tvorbě vzorové aplikace jsem vycházel z ukázkových programů ze stránek LoRaWAN Alliance. Odladění programu do funkční podoby zabralo přibližně měsíc aktivního programování, testování v terénu a měření v laboratořích. Testování odhalilo poměrně široké spektrum programových chyb a hardwarových problémů, které zde nemá ani cenu vypisovat. Dlouhá série testů ale přinesla kýžený výsledek.

Vzorová aplikace umožňuje přenos dat ve všech 3 komunikačních třídách sítě LoRaWAN. Přenos dat je obousměrný, takže můžeme nejen odesílat hodnoty naměřených veličin z koncového zařízení, ale i toto zařízení na dálku řídit. Vzorový program přijímá jednoduchý binární povel, který zapíná a vypíná jednu z připojených LED diod. Opačným směrem se vysílá několik napevno zadaných číselných hodnot, které představují fiktivní naměřené hodnoty. Díky tomu jsem mohl jednoduše ověřit, že při procesu šifrování a vysílání paketu nedochází k chybám a následnému pozměnění odesílané informace.

Během testování bylo navázáno spojení hned s několika různými vysílači v okolí. Nejdelší spojení bylo navázáno na vzdálenost přes 20 km. U tohoto spoje můžeme vidět, že parametr SNR (signal to noise ratio) byl -14,5 dB, což znamená, že přijímaný signál byl 14,5 dB pod úrovní okolního rádiového šumu. Běžně je možné demodulovat signály až do hodnoty -20 dB. I tak ale už bylo spojení značně nestabilní, takže se musel přenos paketu několikrát zopakovat, než se podařilo bráně zachytit zprávu a mému vysílači přijmout potvrzení o doručení paketu.



Obrázek 6.1: Trasa rádiového spoje s bránou rrz-pomnikowa24

Průběh komunikace můžeme vidět v tomto výpisu z konzole LoRaWAN sítě. Je zde vidět, že brána sice korektně přijala paket s pořadovým číslem 2 a odeslala potvrzení o příjmu, jenže toto potvrzení nebylo zachyceno koncovým zařízením, které se tím pádem domnívalo, že byl přenos neúspěšný. Protože bylo ve vzorové aplikaci nastaveno potvrzování příjmu, koncové zařízení opakovalo odesílání 2. paketu tak dlouho, dokud nezachytilo potvrzení o doručení. Jedná se o volitelné nastavení, zařízení může klidně vysílat i bez toho, že by ověřovalo úspěšnost doručení zprávy. Z výpisu můžeme dále vyčíst obsah zprávy v hexadecimálním tvaru. Zpráva byla zakódována pomocí standardu Cayenne LPP, což je jeden z formátů, který usnadňuje navázání webových vizualizací na backend sítě LoRaWAN – v tomto případě servery The Things Network. Zpráva v sobě nese nejen informaci o hodnotě, ale i o datovém typu, takže již není potřeba ve vizualizaci dodatečně nastavovat, jak má interpretovat přijatá data. V tomto případě se jednalo o analogovou hodnotu (číslo v rozsahu 0-255) 11, binární hodnotu 0 a číslo s plovoucí desetinnou čárkou s hodnotou 23,0.

APPLICATION DATA										
<div> <div> <div></div> <div>pause</div> <div></div> <div>clear</div> </div> </div>										
Filters	<div> <div>uplink</div> <div>downlink</div> <div>activation</div> <div>ack</div> <div>error</div> </div>									
	time	counter	port							
	▲ 16:06:03	3	2	confirmed	payload:	00 00 00 01 02 04 4C 02 67 00 E6	analog_in_1:	11	digital_in_0:	0 temperature_2: 23
	▼ 16:06:03		2	confirmed	payload:	01				
	▼ 16:05:37		2	confirmed	payload:	01				
	▲ 16:05:37	2	2	retry confirmed	payload:	00 00 00 01 02 04 4C 02 67 00 E6	analog_in_1:	11	digital_in_0:	0 temperature_2: 23
	▼ 16:05:31		2	confirmed	payload:	01				
	▲ 16:05:31	2	2	retry confirmed	payload:	00 00 00 01 02 04 4C 02 67 00 E6	analog_in_1:	11	digital_in_0:	0 temperature_2: 23
	▲ 16:05:26	2	2	retry confirmed	payload:	00 00 00 01 02 04 4C 02 67 00 E6	analog_in_1:	11	digital_in_0:	0 temperature_2: 23
	▼ 16:05:26		2	confirmed	payload:	01				
	▲ 16:05:21	2	2	retry confirmed	payload:	00 00 00 01 02 04 4C 02 67 00 E6	analog_in_1:	11	digital_in_0:	0 temperature_2: 23
	▼ 16:05:21		2	confirmed	payload:	01				
	▲ 16:05:16	2	2	retry confirmed	payload:	00 00 00 01 02 04 4C 02 67 00 E6	analog_in_1:	11	digital_in_0:	0 temperature_2: 23
	▼ 16:05:16		2	confirmed	payload:	01				

Obrázek 6.2: Výpis komunikace sítě LoRaWAN

Další testy prokázaly, že knihovna umožňuje bezproblémovou autentizaci zařízení do sítě. Po zapnutí zařízení je totiž nejdříve potřeba provést přihlášení zařízení do sítě LoRaWAN. Existují 2 metody. Metoda ABP spočívá v pevném nastavení všech identifikátorů zařízení, ale existují u ní jistá bezpečnostní rizika. Vhodnější metodou pro plošné nasazování je metoda OTAA, kdy při přihlašování zařízení do sítě obdrží koncové zařízení příslušné identifikátory a podrobnosti o nastavení komunikace. Testování ukázalo, že u metody OTAA může při špatném signálu trvat počáteční autentizace i pár minut, než se podaří korektně zachytit všechny potřebné pakety. Komunikace by totiž měla respektovat omezení délky vysílacího času, takže je mezi jednotlivými opakováními přenosu určitá prodleva. Ostatně knihovna si tento interval hlídá sama a nedovolí opakovat přenos dřív, než doběhne vypočtený čekací interval.

Níže uvedený obrázek ukazuje informace o dalším testovacím přenosu na o něco kratší vzdálenost. Díky tomu, že brána odvysílala v metadatech své souřadnice, bylo možné snáze určit její polohu a vzdálenost. V případě, že majitel brány neuvede v metadatech její GPS souřadnice, je možné někdy zjistit pozici brány z jejího identifikátoru. Bohužel se jedná o poměrně pracný postup, protože je potřeba si otevřít mapy na portálu ttnmapper.org a ručně najít v mapě bránu s daným identifikátorem. Portál totiž nemá vyhledávač pro nalezení LoRaWAN brány podle jejího identifikátoru.



Obrázek 6.3: Trasa a výškový profil rádiového spoje

Protože byla přenosová vzdálenost poměrně malá, bylo spojení stabilní, takže nebylo potřeba opakovat žádný paket se zprávou. Na začátku výpisu můžeme vidět, že se zařízení přihlásilo do sítě. Následují dva přijaté pakety s daty a k nim příslušné potvrzení přijetí zprávy.

APPLICATION DATA

pause

clear

Filters

uplink

downlink

activation

ack

error

	time	counter	port									
▼	15:10:48		0									
▲	15:10:48	2	2	confirmed	payload:	00 00 00 01 02 04 4C 02 67 00 E6	analog_in_1:	11	digital_in_0:	0	temperature_2:	23
▼	15:10:24		0									
▲	15:10:24	1	2	confirmed	payload:	00 00 00 01 02 04 4C 02 67 00 E6	analog_in_1:	11	digital_in_0:	0	temperature_2:	23
⚡	15:10:16	dev addr: 26 01 27 FF app eui: 70 B3 D5 7E D0 03 E5 87 dev eui: 00 11 2F 9A 68 1A F2 29										

Obrázek 6.4: Výpis dat z backendu LoRaWAN

Kapitola 7

Zhodnocení výsledků

Na základě výsledků předběžných testů knihovny mohu prohlásit, že byla praktická část práce úspěšná. V průběhu vývoje jsem měl sice pochybnosti o proveditelnosti tak rozsáhlých úprav softwaru, ale ukázalo se, že je tato úprava skutečně proveditelná. Podařilo se úspěšně překlénout hardwarové rozdíly mezi portovanou knihovnou a mikrokontrolérem KL25Z. Dokonce se podařilo vyhnout úpravám vyšších vrstev knihovny a provádět pouze modifikace na nejnižší hardwarové úrovni. Domnívám se tedy, že by mělo být možné přenést vytvořenou hardwarovou abstrakční vrstvu do novějších verzí oficiální knihovny. Během doby, kdy jsem pracoval na tomto softwaru, totiž zároveň pokročila standardizace LoRaWAN protokolu a byla dokončena nová verze knihovny LoRa MAC Node. Možná by stálo za pokus sloučit tento projekt s nejnovější verzí knihovny a získat tak podporu novějšího komunikačního standardu.

Z časových důvodů již nebylo možné provést návrh komplexnější aplikace s využitím celé funkcionality knihovny. Především by bylo potřeba provést otestování funkcionality knihovny v kombinaci s různými spánkovými režimy mikrokontroléru. Ideální stav by byl, kdyby byla tato rozsáhlejší aplikace podrobena dlouhodobějšímu testu. Tím by se skutečně ověřila spolehlivost a bezchybnost portované knihovny. Rozsah provedených testů prokázal, že je portovaná knihovna provozuschopná a je možné na ní založit vývoj dalšího softwaru pro vestavěné zařízení s mikrokontrolérem KL25Z. Nemohu ale s jistotou prohlásit, že je knihovna jako celek napsaná bezchybně – rozsah testů nad celou knihovnou byl v tomto ohledu skutečně omezený. Knihovna má poměrně široké možnosti parametrizace, ať už v oblasti možností zabezpečení, různých režimů komunikace, vytváření různých front pro pakety a zachytávání různých událostí. Důkladné otestování se nejspíš neobejde bez řádného návrhu testovacích scénářů, aby byly pokryty alespoň běžné kombinace nastavení.

S jistotou jsou plně funkční následující funkce:

- Obousměrná komunikace ve třídě A.
- Obousměrná komunikace ve třídě C.
- Přepínání komunikačních tříd za chodu zařízení.

- Autorizace zařízení pomocí ABP.
- Autorizace zařízení pomocí OTAA.
- Vysílání v rádiovém pásmu EU868.
- Změny nastavení modemu (vysílací výkon, modulace, volba zdroje napájení atd.).

Komunikaci ve třídě B se nepodařilo navázat. Program sice obsahuje všechny potřebné náležitosti pro uskutečnění tohoto přenosu, nicméně testy v terénu ukázaly, že se zde vyskytuje neznámý problém. Zařízení se úspěšně zaregistrovalo do sítě a začalo čekat na synchronizační signál od brány LoRaWAN. Tento signál je vysílán každé 2 minuty. Bohužel se jej nikdy nepodařilo zachytit, takže testované zařízení si nemohlo srovnat vnitřní hodiny s časem rádiové sítě a naplánovat další přenos paketů.

Jako největší nedostatek tohoto projektu vidím nedostatečnou dokumentaci celého softwaru. Oficiální dokumentace neříká vůbec nic o detailech fungování vnitřních částí knihovny. Omezuje se pouze na stručné výčty definovaných metod a datových typů a pár odstavců o tom, co je potřeba implementovat v případě portace knihovny. Jako návod k použití slouží několik ukázek vzorových aplikací a to je vše. Z vlastní zkušenosti vím, že dokonce ani některé ukázky implementace knihovny nebyly funkční v kombinaci s portovanou knihovnou. Možná ale byla chyba jen v nastavení mého kompilátoru nebo konfiguraci knihovny. V samém důsledku programátorovi nezbyvá nic jiného, než prohledávat zdrojové kódy fulltextovým vyhledávačem a domýšlet si, co bylo záměrem autora. Zde ještě mohu doporučit porovnávat terminologii a zkratky v kódu s dokumentem standardu příslušné verze LoRaWAN. Autoři softwaru dodržují terminologii zavedenou ve specifikačním dokumentu, z čehož je možné si odvodit chování některých částí programu. Snažil jsem se doplnit nejasné části kódu komentáři a odkazy na související části kódu, ale ani tak není knihovna stále dostatečně zdokumentována.

Kapitola 8

Závěr

V této práci bylo shrnuto téma nízkoenergetických zařízení. Dále byl vytvořen souhrn informací o LPWAN z pohledu technických parametrů a specifik jednotlivých řešení. U některých technologií je trochu problém získat přesné konkrétní informace o parametrech, protože je značná část materiálů spíše reklamního charakteru. Z rozboru softwarových knihoven pro LoRaWAN vyplynul trochu neutěšený stav okolo vývoje softwarů pro tuto technologii. Na druhou stranu je to asi dáno i rychlým vývojem, a s tím spojenými častými změnami. Asi jediným dlouhodoběji stabilním projektem je knihovna LoRa MAC Node, z které vychází tato diplomová práce. Rozhodnutí zvolit právě tuto knihovnu se ukázalo jako velmi vhodné.

Úprava knihovny pro vývojový kit FRDM-KL25Z se z počátku jevila jako poměrně jednoduché a schůdné řešení. Na druhou stranu zde nebyla žádná jistota, že se úprava softwaru podaří. Skutečná náročnost celé operace se ukázala až v průběhu práce, nicméně naštěstí nebyla nalezena žádná překážka, která by znemožnila dokončení portace knihovny na vybranou platformu. Portaci tedy můžeme považovat za úspěšnou. Dílčí jednotkové testy knihovny dopadly úspěšně, všechny testované periferie fungují podle očekávání. Z časových důvodů už nebyl prostor na provedení důkladných testů celé knihovny, takže bylo provedeno jen asi 6 rádiových přenosů při různých kombinacích nastavení. Počet možných konfigurací knihovny je ale zdaleka vyšší.

Hotová knihovna je připravena pro použití, ale vřele doporučuji provést ještě další důkladné testování. Jako další krok bych zvolil vytvoření komplexnějšího programu, který by využil celý potenciál této knihovny. Zejména bych se zaměřil na použití spánkových režimů mikrokontroléru a optimalizaci energetické náročnosti, která je klíčovým bodem ke konstrukci nízkoenergetického zařízení.

Z mého pohledu se jednalo o můj první seriózní projekt na 32bitovém mikrokontroléru. V době zahájení práce jsem měl pouze základní znalosti práce s touto platformou, takže jsem se spíše učil za pochodu. Na druhou stranu to byl pro mě celkem velký skok kupředu, protože jsem do té doby pracoval převážně s 8bitovými mikrokontroléry.

Literatura

1. *Claude Chappe* [online]. 2021-01-19 [cit. 2021-02-02]. Dostupné z: <https://www.britannica.com/biography/Claude-Chappe>.
2. *8-bit Low-Power PIC® and AVR® Microcontrollers* [online] [cit. 2021-04-10]. Dostupné z: <https://www.microchip.com/en-us/solutions/low-power/8-bit-low-power-mcus>.
3. *32-bit Low-Power PIC® and SAM Microcontrollers* [online] [cit. 2021-04-10]. Dostupné z: <https://www.microchip.com/en-us/solutions/low-power/32-bit-ultra-low-power-and-low-power-mcus>.
4. GOLDBERG, Lee H. *Developing Energy Budgets for Low-Power Sensing Systems - Part II: The Art of the Power Budget* [online]. 2012-02-29 [cit. 2021-04-11]. Dostupné z: <https://www.digikey.com/en/articles/developing-energy-budgets-for-low-power-sensing-systems---part-ii-the-art-of-the-power-budget>.
5. *TMP102 Low-Power Digital Temperature Sensor With SMBus and Two-Wire Serial Interface in SOT563* [online] [cit. 2021-04-10]. Dostupné z: <https://www.ti.com/lit/ds/symlink/tmp102.pdf>.
6. SCHACKMANN, Ren. *Low-Power Design Techniques for Temperature-Sensing Applications* [online] [cit. 2021-04-10]. Dostupné z: <https://www.ti.com/lit/an/snua026/snua026.pdf?ts=1618389858734>.
7. *BU-106: Advantages of Primary Batteries* [online] [cit. 2021-04-16]. Dostupné z: https://batteryuniversity.com/learn/article/primary_batteries.
8. *BU-209: How does a Supercapacitor Work?* [Online] [cit. 2021-04-16]. Dostupné z: https://batteryuniversity.com/learn/article/whats_the_role_of_the_supercapacitor.
9. HRZINA, Pavel. *Cykly a životnost baterie* [online]. 2020-09-01 [cit. 2021-04-16]. Dostupné z: <https://oze.tzb-info.cz/akumulace-elektriny/21096-cykly-a-zivotnost-baterie>.
10. KU, Meng-Lin; LI, Wei; CHEN, Yan; RAY LIU, K. J. Advances in Energy Harvesting Communications: Past, Present, and Future Challenges. *IEEE Communications Surveys Tutorials*. 2016, roč. 18, č. 2, s. 1384–1412. Dostupné z DOI: 10.1109/COMST.2015.2497324.

11. HUANG, Kaibin; ZHOU, Xiangyun. Cutting the last wires for mobile communications by microwave power transfer. *IEEE Communications Magazine*. 2015, roč. 53, č. 6, s. 86–93. Dostupné z DOI: 10.1109/MCOM.2015.7120022.
12. HOLMES, A.S.; HONG, G.; PULLEN, K.R.; BUFFARD, K.R. Axial-flow microturbine with electromagnetic generator: design, CFD simulation, and prototype demonstration. In: *17th IEEE International Conference on Micro Electro Mechanical Systems. Maastricht MEMS 2004 Technical Digest*. 2004, s. 568–571. Dostupné z DOI: 10.1109/MEMS.2004.1290648.
13. PARADISO, J.A.; STARNER, T. Energy scavenging for mobile and wireless electronics. *IEEE Pervasive Computing*. 2005, roč. 4, č. 1, s. 22. Dostupné z DOI: 10.1109/MPRV.2005.9.
14. *Geolocation technologies* [online] [cit. 2021-02-20]. Dostupné z: <https://build.sigfox.com/geolocation-technologies>.
15. *Certifying LoRaWAN Products* [online] [cit. 2021-03-04]. Dostupné z: <https://loro-alliance.org/lorawan-certification/>.
16. TELKAMP, Thomas; SLATS, Laurens. *Ground breaking world record! LoRaWAN packet received at 702 km (436 miles) distance* [online]. 2017-09-08 [cit. 2021-02-27]. Dostupné z: <https://www.thethingsnetwork.org/article/ground-breaking-world-record-lorawan-packet-received-at-702-km-436-miles-distance>.
17. *What are LoRa and LoRaWAN?* [Online] [cit. 2021-03-09]. Dostupné z: <https://loro-developers.semtech.com/library/tech-papers-and-guides/lora-and-lorawan/>.
18. *Mapa TTN* [online] [cit. 2021-03-09]. Dostupné z: <https://ttnmapper.org/alpha-shapes/>.
19. *Mapa Českých radiokomunikací* [online] [cit. 2021-03-09]. Dostupné z: <https://www.iot-portal.cz/mapa-pokryti/>.
20. *IQRF Technical Guide* [online] [cit. 2021-03-15]. Dostupné z: https://static.iqrf.org/Technical-guide_IQRF_181020.pdf.
21. *IQRF OS Operating System - User's Guide* [online] [cit. 2021-03-15]. Dostupné z: https://static.iqrf.org/User_Guide_IQRF-OS-404D_TR-7xD_200918.pdf.
22. *Discovery: Automated creating a routing structure* [online] [cit. 2021-03-21]. Dostupné z: <https://www.iqrf.org/technology/iqmesh/discovery>.
23. *Operating system: Two-layer IQRF architecture* [online] [cit. 2021-03-25]. Dostupné z: <https://www.iqrf.org/technology/os>.
24. *DPA: Three-layer IQRF architecture* [online] [cit. 2021-03-25]. Dostupné z: <https://www.iqrf.org/technology/dpa>.
25. *SigFox story* [online] [cit. 2021-02-20]. Dostupné z: <https://www.sigfox.com/en/sigfox-story>.

26. *SigFox coverage* [online] [cit. 2021-02-20]. Dostupné z: <https://www.sigfox.com/en/coverage>.
27. *Je roaming opravdu slabinou NB-IoT?* [Online]. 2020-03-25 [cit. 2021-02-27]. Dostupné z: <https://www.iot-portal.cz/2020/03/25/je-roaming-opravdu-slabinou-nb-iot/>.
28. PECH, Jiří. *IOT TECHNOLOGIE: SIGFOX* [online]. 2019-05-09 [cit. 2021-02-27]. Dostupné z: <https://www.eman.cz/blog/iot-technologie-sigfox-4-5/>.
29. *SigFox for M2M and IoT* [online] [cit. 2021-02-20]. Dostupné z: <https://www.electronics-notes.com/articles/connectivity/sigfox/what-is-sigfox-basics-m2m-iot.php>.
30. *What is Atlas?* [Online] [cit. 2021-02-20]. Dostupné z: <https://build.sigfox.com/geolocation-sigfox-atlas>.
31. *Sigfox Atlas* [online] [cit. 2021-02-27]. Dostupné z: <https://build.sigfox.com/geolocation-sigfox-atlas>.
32. *Základní ceník konektivity* [online] [cit. 2021-03-02]. Dostupné z: <https://sigfox.cz/cs/onas/cenik-vop>.
33. MEKKI, Kais; BAJIC, Eddy; CHAXEL, Frederic; MEYER, Fernand. A comparative study of LPWAN technologies for large-scale IoT deployment. *ICT Express*. 2017. Dostupné z DOI: 10.1016/j.icte.2017.12.005.
34. *NB-IoT explained: a complete guide to Narrowband-IoT* [online]. 2021-01-29 [cit. 2021-03-25]. Dostupné z: <https://www.i-scoop.eu/internet-of-things-guide/lpwan/nb-iot-narrowband-iot/>.
35. *Mobile IoT Deployment Map* [online]. 2020-12-01 [cit. 2021-03-20]. Dostupné z: <https://www.gsma.com/iot/deployment-map>.
36. *LTE-M* [online]. 2018-02-16 [cit. 2021-03-20]. Dostupné z: <https://www.u-blox.com/en/technologies/lte-cat-m>.
37. *Overview of LoRaWAN Libraries [HowTo]* [online]. 2019 [cit. 2020-11-15]. Dostupné z: <https://www.thethingsnetwork.org/forum/t/overview-of-lorawan-libraries-howto/24692/2>.
38. *Arduino-LMIC library* [online]. 2005-10-16 [cit. 2020-11-15]. Dostupné z: <https://github.com/matthijskooijman/arduino-lmic>.
39. *RP2-1.0.2 LoRaWAN® Regional Parameters* [online] [cit. 2020-11-16]. Dostupné z: <https://loro-alliance.org/resource-hub/rp2-102-lorawan-regional-parameters>.
40. *Arduino-LMIC* [online]. 2019 [cit. 2020-11-16]. Dostupné z: <https://github.com/mcci-catena/arduino-lmic>.

41. *KL25 Sub-Family Reference Manual*. 2012-09-03. Dostupné také z: <https://www.nxp.com/docs/en/reference-manual/KL25P80M48SF0RM.pdf>.
42. *FRDM-KL25Z User's Manual*. 2013-10-24. Dostupné také z: <https://www.mouser.com/pdfdocs/FRDM-KL25Z.pdf>.